NSS
Neutrosophic Sets and Systems, Vol. 87, 2025
University of New Mexico

Improving the Routing Security in Wireless Sensor Networks using Neutrosophic Set and Machine Learning Models

Hanadi Ahmad Simmak[1], Ahmed A El-Douh[2,3,7], Tareef S Alkellezli[4], Rabih Sbera[5], Darin shafek[5], Ahmed Abdelhafeez[6,7]

[1]Biomedical Engineering Department, Collage of Engineering, Ashur University, Baghdad, Iraq
[2]Information Systems Department, Faculty of Information Systems and Computer Science, October 6 University, Giza, 12585, Egypt
[3]School of Cyber Science and Engineering, Huazhong University of Science and Technology, 1037, Hongshan, Wuhan 430074, China
[4]Faculty of Informatics Engineering, Department of Computer Systems and Networks Engineering, University of Homs, Homs, Syria
[5]College of Mechanical and Electrical Engineering, Computer and Automatic Control Engineering Department, Lattakia University, Lattakia, Syria
[6]Computer Science Department, Faculty of Information Systems and Computer Science, October 6 University, Giza, 12585, Egypt
[7]Applied Science Research Center, Applied Science Private University, Amman

Abstract: Numerous methods have been put forth to identify and safeguard routing data because Wireless Sensor Networks (WSNs) are susceptible to attacks during data transfer. To create an artificial intelligence-based attack detection system for WSNs, we provide a unique stochastic predictive machine learning technique in this research that is intended to identify unreliable events and untrustworthy routing properties. Our approach makes use of real-time feature analysis of simulated WSN routing data. We create a strong foundation for categorization. Our approach's primary benefit is the development of an effective machine learning (ML) technique that can analyze and filter WSN traffic to stop dangerous and suspicious data, lessen the significant variation in the routing information gathered, and quickly identify assaults before they happen. We use the XGBoost and Random Forest (RF) models with different parameters. Then the bipolar neutrosophic set is used to deal with uncertainty and vague information. The neutrosophic set is used to rank the ML models and select the best one.

Keywords: Bipolar Neutrosophic Numbers; Uncertainty; Wireless Sensor Networks; Security. Attacks.

1. Introduction

Most wireless sensor networks (WSNs) are made up of several tiny sensor nodes that are battery-operated. These sensors are set up to gather information and exchange messages with one another using a routing protocol, which enables them to send the data they have gathered throughout the

network. Malicious threats that impact network resource availability may surface during the forwarding process[1], [2]. One of the main issues with WSNs is energy resource conservation.

Consequently, network transmission, latency, throughput, energy consumption, and network longevity are some of the characteristics that limit WSN performance[3], [4]. There are several important factors that impact WSN performance, including throughput, which is the maximum number of packets that the nodes can send, energy consumption, which is the amount of energy consumed by the nodes during the routing phase, and lifetime, which is the total amount of time that the nodes in the network remain alive until the first node uses up all of its energy.

The implementation of secure route detection models is necessary for effectively protecting and preserving energy resources inside a network. These models are created by carefully examining large amounts of route data. The inherent constraints of WSNs may be addressed via an adaptive secure routing system.[5], [6]. Such frameworks improve performance indicators and strengthen defenses against possible security breaches by reducing excessive network activity and node workloads. In this paper, we present a novel machine learning system that is distinguished by its stochastic nature and intelligence. The WSN routing feature dataset may be thoroughly analyzed and filtered thanks to this approach. ML models are applied in different applications.[7], [8].

A single expert cannot address complex real-life problems, and using straightforward techniques and instruments in decision-making procedures is insufficient in many situations.[9], [10]. When addressing ambiguity and uncertainty in these issues, a broad spectrum of knowledge must be included.

The membership function, sometimes referred to as the truth function, the non-membership or falsehood function, and the indeterminacy function, on the other hand, are the three mutually independent functions that form the basis of Smarandache's neutrosophic set theory. This makes it possible to represent even the most ambiguous and unclear situations.[11], [12]. Fuzzy set theory, intuitionistic fuzzy sets, grey sets, and vague sets based on neutrosophy are all essentially generalized in neutrosophic set theory.

Fuzziness, neutrosophic, greyness, and vagueness are characteristics of decision-making processes. To meet the needs of contemporary society, they must be resilient and strategic rather than static, and they must change in response to unpredictable and hybrid situations.[13], [14]. It is challenging to create models for real-world issues without a deep understanding of how people make decisions. Because of this, it is crucial to include the hybrid approach of neutrosophic sets into the idea of decision-making.

2. Neutrosophic Set

This section shows the definitions of bipolar neutrosophic sets (BNSs) [15] And the steps of the ML Models to detect attacks.

$$U = \left\{ Q, \left(T_U^+(Q), I_U^+(Q), F_U^+(Q), T_U^-(Q), I_Q^-(U), F_Q^-(U) \right) u \in U \right\} \quad (1)$$

$$T_U^+(Q), I_U^+(Q), F_U^+(Q): U \to [0,1] \tag{2}$$

$$T_U^-(Q), I_U^-(Q), F_U^-(Q): U \to [-1,0] \tag{3}$$

We show operations of two bipolar neutrosophic numbers (BNNs) such as:

$$U_1 = \{T_1^+(Q), I_1^+(Q), F_1^+(Q), T_1^-(Q), I_1^-(Q), F_1^-(Q)\}, U_2 = \{T_2^+(Q), I_2^+(Q), F_2^+(Q), T_2^-(Q), I_2^-(Q), F_2^-(Q)\}$$

$$U_1 \cup U_2 = \begin{pmatrix} \max\left(T_1^+(Q), T_2^+(Q)\right), \frac{I_1^+(Q)+I_2^+(Q)}{2}, \\ \min\left(F_1^+(Q), F_2^+(Q)\right), \min\left(T_1^-(Q), T_2^-(Q)\right), \frac{I_1^-(Q)+I_2^-(Q)}{2}, \\ \max\left(F_1^-(Q), F_2^-(Q)\right) \end{pmatrix} \tag{4}$$

$$U_1 + U_2 = \begin{pmatrix} T_1^+(Q) + T_2^+(Q) - T_1^+(Q)T_2^+(Q), \\ I_1^+(Q)I_2^+(Q), \\ F_1^+(Q)F_2^+(Q), \\ -T_1^-(Q)T_2^-(Q), \\ -\left(-I_1^-(Q) - I_2^-(Q) - I_1^-(Q)I_2^-(Q)\right), \\ -\left(-F_1^-(Q) - F_2^-(Q) - F_1^-(Q)F_2^-(Q)\right) \end{pmatrix} \tag{5}$$

$$U_1U_2 = \begin{pmatrix} T_1^+(Q)T_2^+(Q), I_1^+(Q) + I_2^+(Q) - I_1^+(Q)I_2^+(Q) + \\ F_1^+(Q) + F_2^+(Q) - F_1^+(Q)F_2^+(Q), \\ -\left(-T_1^-(Q) - T_2^-(Q) - T_1^-(Q)T_2^-(Q)\right), \\ -I_1^-(Q)I_2^-(Q), \\ -F_1^-(Q)F_2^-(Q) \end{pmatrix} \tag{6}$$

$$KU_1 = \begin{pmatrix} \left(1 - \left(1 - T_1^+(Q)\right)\right)^K, \\ \left(I_1^+(Q)\right)^K, \\ \left(F_1^+(Q)\right)^K, \\ -\left(-\left(T_1^-(Q)\right)^K\right), \\ -\left(-\left(I_1^-(Q)\right)^K\right), \\ -\left(1 - \left(1 - F_1^-(Q)\right)\right)^K \end{pmatrix} \tag{7}$$

$$U_1^K = \begin{pmatrix} \left(T_1^+(Q)\right)^K, \\ \left(1 - \left(1 - I_1^+(Q)\right)\right)^K, \\ \left(1 - \left(1 - F_1^+(Q)\right)\right)^K, \\ -\left(1 - \left(1 - T_1^-(Q)\right)\right)^K, \\ -\left(-\left(I_1^-(Q)\right)^K\right) \\ -\left(-\left(F_1^-(Q)\right)^K\right) \end{pmatrix} \tag{8}$$

3. Results and Discussion

We utilized a simulation dataset to put the suggested attack detection methods into practice. Network Simulator version 2 (ns-2.35) was used to gather data in clustered sample network situations. A cluster head was chosen after 14 rounds before the node lifespan began to decline and perish. There were 100 nodes in the network overall.

We used an existing dataset model named WSN-DS to evaluate the suggested system and choose the top classifiers. The collection of necessary sensor data in the dataset is made up of both sent and received packets in a WSN using a tracking technique. Five simulation scenarios' worth of entries make up the dataset. We set aside 80% of the data to construct classifier models, with the remaining 20% being utilized for testing, to maximize performance during training using dimensionality reduction.

The WSN-DS wireless sensor network open-access dataset [12], created with the hierarchical routing system LEACH, is the dataset utilized in this study. To determine the behavior of sensor nodes in the WSN, 23 characteristics were taken from the routing scenario. Over 374,661 entries of Wireless Sensor Network (WSN) behavior, divided into five kinds (Normal, Blackhole, Grayhole, and Flooding) as shown in Figure 1, are among the attributes included in the collection. Our suggested models are trained using the routing characteristics that were gathered using a popular clustered routing protocol. Figure 2 shows the heatmap. Figure 3 shows the violin plot. Figure 4 shows the strip plot.

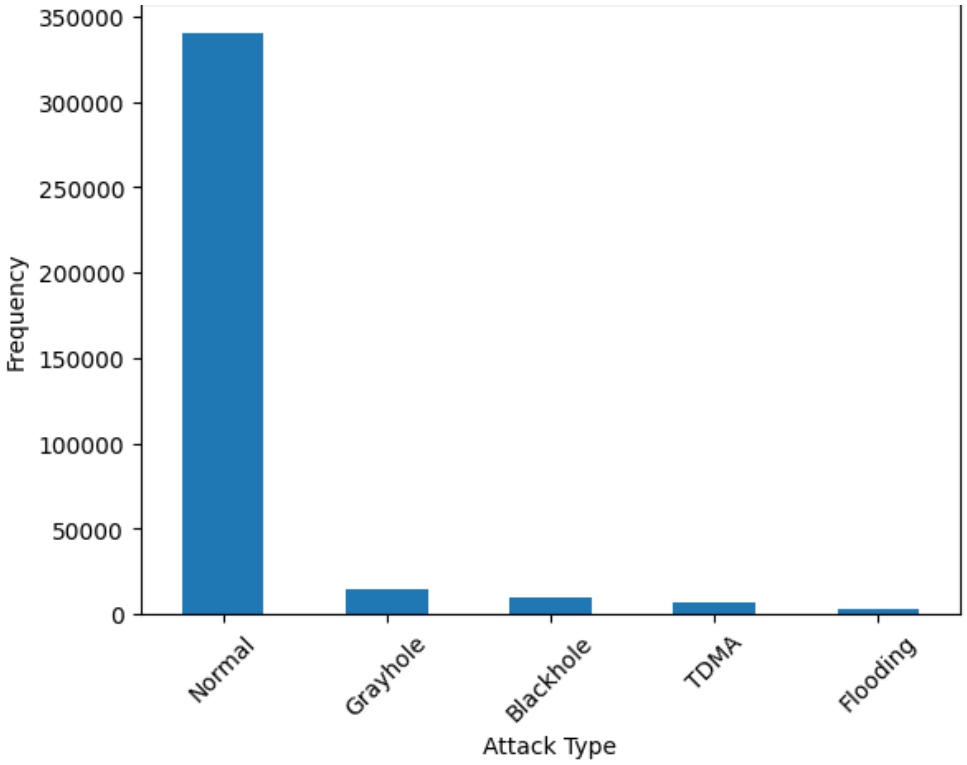

Figure 1. Five classes.

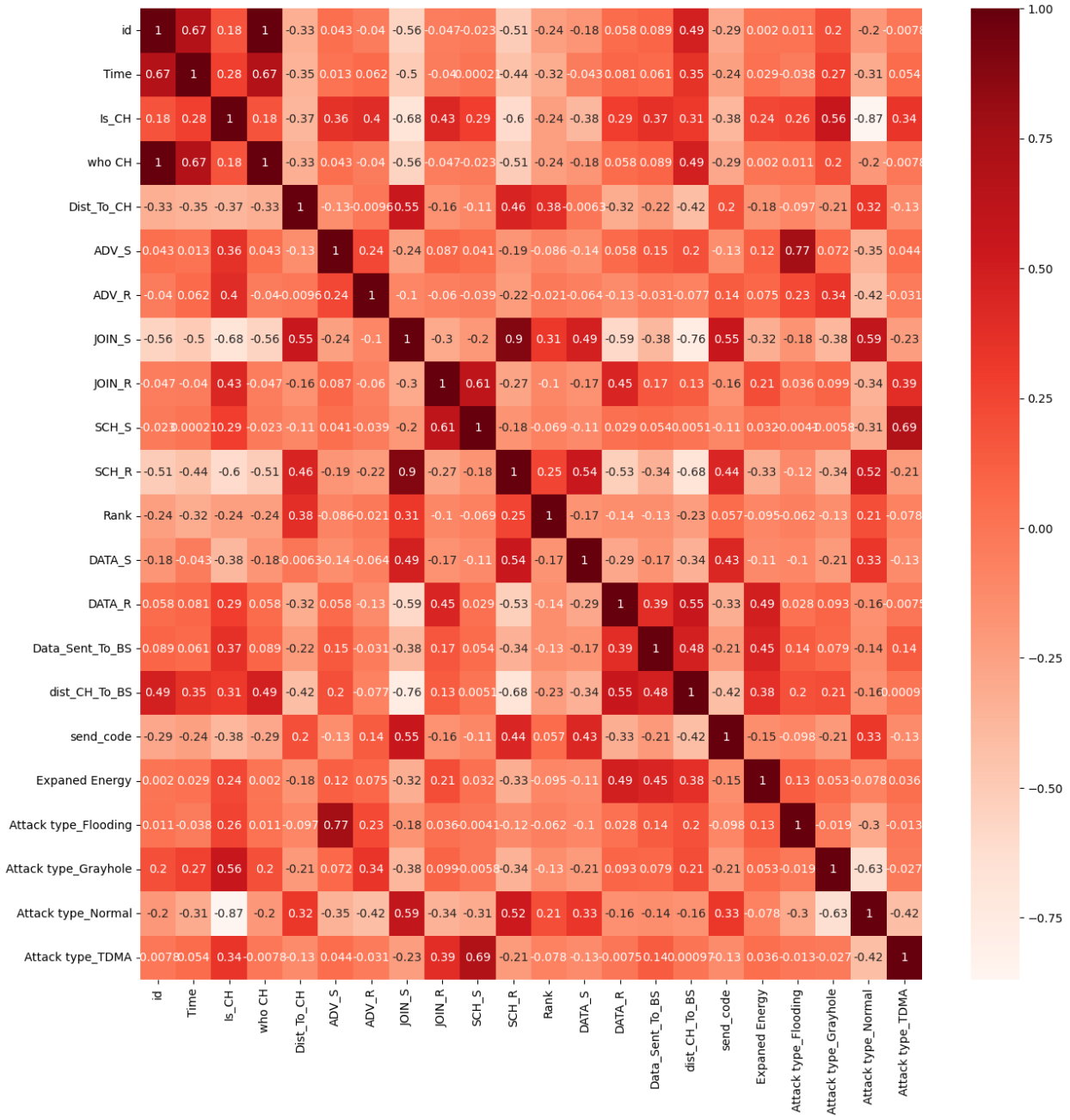

Figure 2. Heatmap.

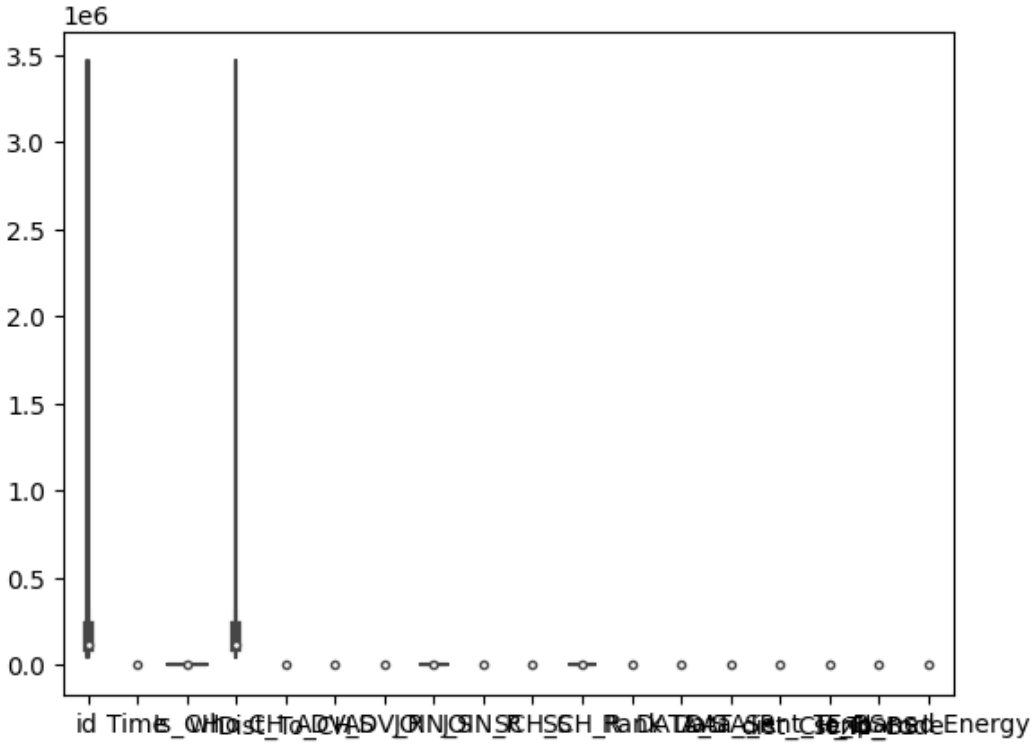

Figure 3. The violin plot.

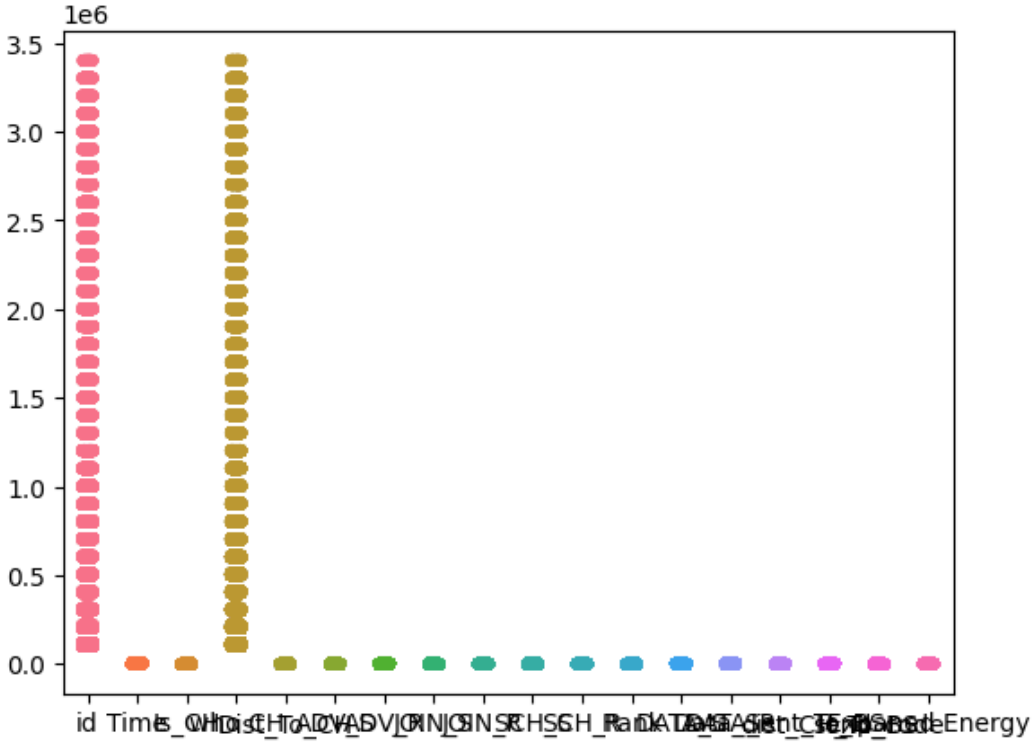

Figure 4. The stripplot.

Table 1 shows the evaluation matrices for random forest and XGBoost with 200,150,100,50,25 estimators.

Table 1. ML Models.

	Accuracy	Precision	Recall	F1-score
XGBoost with estimator 100	0.99712	0.997132	0.997117	0.9971
XGBoost with estimator 50	0.99666	0.99669	0.996664	0.996649
RF	0.997037	0.997071	0.997037	0.997026
XGBoost with estimator 200	0.99722	0.99724	0.997224	0.997207
XGBoost with estimator 150	0.99717	0.99718	0.997171	0.997154
XGBoost with estimator 25	0.9958	0.99583	0.995796	0.995781

4. Neutrosophic Model Analysis

Then we show the results of the neutrosophic model to select the best ML model. We show the steps of the neutrosophic model as:

Step 1. Build the decision matrix.

Step 2. Compute the criteria weights.

Steps 3. Compute the weighted decision matrix.

Step 4. Rank the alternatives.

In the first step, six alternatives are created in the decision matrix as shown in Table 2. They use the bipolar neutrosophic numbers to evaluate the criteria and alternatives.

Table 2. Decision matrix.

	$WSNC_1$	$WSNC_2$	$WSNC_3$	$WSNC_4$
$WSNA_1$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)
$WSNA_2$	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)
$WSNA_3$	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)
$WSNA_4$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.4,0.1,0.4,-0.1,-0.2,-0.5)	(0.4,0.1,0.4,-0.1,-0.2,-0.5)
$WSNA_5$	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)
$WSNA_6$	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)
	$WSNC_1$	$WSNC_2$	$WSNC_3$	$WSNC_4$
$WSNA_1$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)
$WSNA_2$	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)
$WSNA_3$	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)
$WSNA_4$	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.4,0.1,0.4,-0.1,-0.2,-0.5)
$WSNA_5$	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)
$WSNA_6$	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)
	$WSNC_1$	$WSNC_2$	$WSNC_3$	$WSNC_4$
$WSNA_1$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)
$WSNA_2$	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)
$WSNA_3$	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)
$WSNA_4$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)
$WSNA_5$	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)
$WSNA_6$	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)
	$WSNC_1$	$WSNC_2$	$WSNC_3$	$WSNC_4$
$WSNA_1$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)
$WSNA_2$	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)
$WSNA_3$	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)

$WSNA_4$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.7,0.3,0.2,-0.4,-0.2,-0.1)
$WSNA_5$	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)
$WSNA_6$	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)
	$WSNC_1$	$WSNC_2$	$WSNC_3$	$WSNC_4$
$WSNA_1$	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)
$WSNA_2$	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)
$WSNA_3$	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)
$WSNA_4$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)
$WSNA_5$	(0.7,0.3,0.2,-0.4,-0.2,-0.1)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)
$WSNA_6$	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)
	$WSNC_1$	$WSNC_2$	$WSNC_3$	$WSNC_4$
$WSNA_1$	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)
$WSNA_2$	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)
$WSNA_3$	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)
$WSNA_4$	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)
$WSNA_5$	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)	(0.4,0.3,0.3,-0.1,-0.2,-0.3)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)
$WSNA_6$	(0.6,0.4,0.4,-0.3,-0.2,-0.3)	(0.1,0.4,0.3,-0.1,-0.2,-0.3)	(0.1,0.4,0.4,-0.1,-0.3,-0.5)	(0.5,0.4,0.3,-0.4,-0.3,-0.3)

In the second step, we show the criteria weights using the average method. The criteria weights are: 0.252648549, 0.243666513, 0.251957623, 0.251727315.

In the third step, we obtain the weighted decision matrix as shown in Table 3.

Table 3. Weighted decision matrix.

	$WSNC_1$	$WSNC_2$	$WSNC_3$	$WSNC_4$
$WSNA_1$	0.791632	0.731	0.755873	0.721618
$WSNA_2$	0.757946	0.698511	0.747474	0.750986
$WSNA_3$	0.741102	0.714755	0.764271	0.767768
$WSNA_4$	0.791632	0.735061	0.781069	0.809723
$WSNA_5$	0.779	0.735061	0.789467	0.780355
$WSNA_6$	0.757946	0.682266	0.755873	0.755182

In the fourth step, we compute the sum of each row. Then we rank the alternatives as shown in Figure 5. The results show alternative 4 is the best and alternative 6 is the worst. The results of XGBoost, with 200 estimators, are the best.

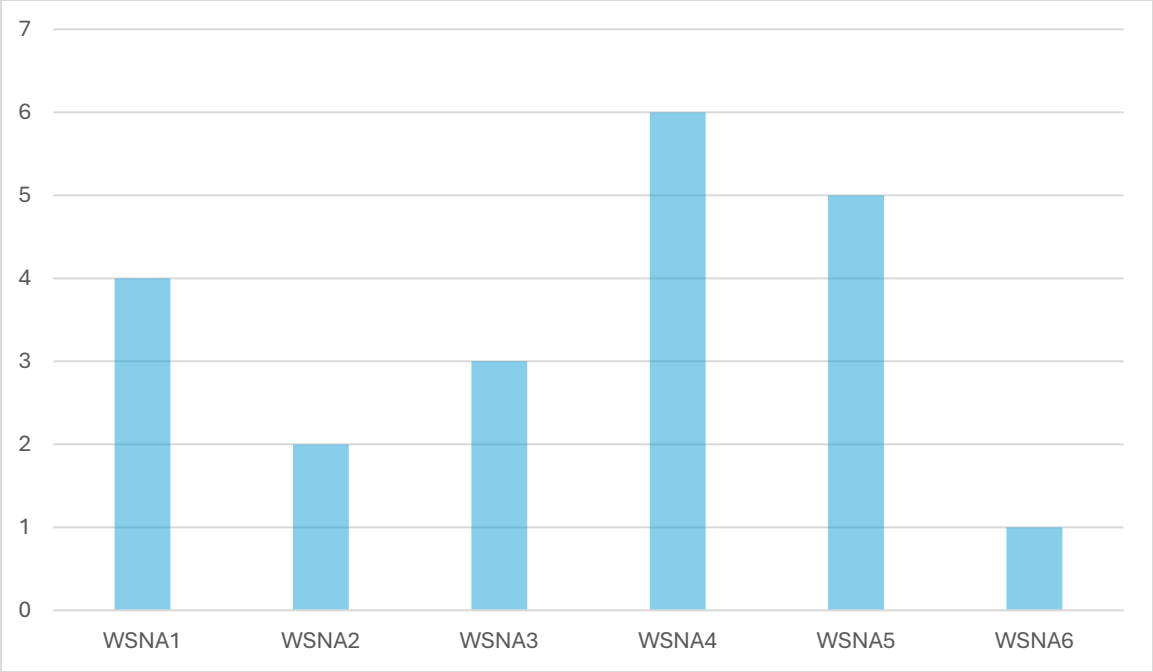

Figure 5. Ranks of alternatives.

5. Conclusions

To identify incorrect data in the wireless sensor networks (WSNs) dataset, we suggested a novel attack detection technique in this study. According to the testing findings, data classification performance may reach 99.72%, 99.71%, and 99.71%. The effectiveness of our strategy might yet be enhanced, though. We used the bipolar neutrosophic set (BNS) to overcome uncertainty and vague information. The neutrosophic model is used to select the best ML model based on evaluation matrices. The results show that the XGBoost with estimator 200 is the best model.

The utilization of high-speed computer servers to analyze the entire dataset without dimensionality reduction techniques is something we intend to investigate in future work and compare the outcomes with the current study. Additionally, we want to present another detection system that can learn from real-time WSNs and anticipate potential assaults or harmful behaviors using deep learning (DL) and artificial neural networks (ANN).

References

[1] A.-S. K. Pathan, H.-W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," in *2006 8th International Conference on Advanced Communication Technology*, IEEE, 2006, pp. 6- pp.

[2] M. Ismail and M. Y. Sanavullah, "Security topology in wireless sensor networks with routing optimisation," in *2008 Fourth International Conference on Wireless Communication and Sensor Networks*, IEEE, 2008, pp. 7–15.

[3] J. Sen, W. Seah, and Y. K. Tan, "Routing security issues in wireless sensor networks: attacks

and defenses," *Sustain. Wireless. Sens. Networks*, pp. 279–309, 2010.

[4] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2–3, pp. 293–315, 2003.

[5] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[6] V. C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in wireless sensor networks," *Wirel. Commun. Mob. Comput.*, vol. 8, no. 1, pp. 1–24, 2008.

[7] M. R. Abdellah, R. E. Owaidah, S. M. Selem, and A. H. Abdel-aziem, "Revisiting Machine Learning for Predictive Modeling for Stroke from Electronic Health Records," *SciNexuses*, vol. 1, pp. 16–27, 2024.

[8] A. Abdelhafeez, A. Ashraf, and H. Elbehiery, "Harnessing Machine Learning for Accurate Cardiovascular Disease Prediction," *SciNexuses*, vol. 1, pp. 9–15, 2024.

[9] V. Christianto and F. Smarandache, "Neutrosophic Logic Guide to Risk Management Especially Given Stable Pareto Distribution," *SciNexuses*, vol. 2, pp. 27–32, 2025.

[10] T. Fujita, "Note for neutrosophic incidence and threshold graph," *SciNexuses*, vol. 1, pp. 97–125, 2024.

[11] F. Smarandache and M. Jdid, "An Overview of Neutrosophic and Plithogenic Theories and Applications," 2023.

[12] F. Smarandache, *Neutrosophic precalculus and neutrosophic calculus: neutrosophic applications*. Infinite Study, 2015.

[13] M. Ali, L. H. Son, I. Deli, and N. D. Tien, "Bipolar neutrosophic soft sets and applications in decision making," *J. Intell. Fuzzy Syst.*, vol. 33, no. 6, pp. 4077–4087, 2017.

[14] I. Deli, M. Ali, and F. Smarandache, "Bipolar neutrosophic sets and their application based on multi-criteria decision making problems," in *2015 International conference on advanced mechatronic systems (ICAMechS)*, IEEE, 2015, pp. 249–254.

[15] V. Ulucay, I. Deli, and M. Şahin, "Similarity measures of bipolar neutrosophic sets and their application to multiple criteria decision making," *Neural Comput. Appl.*, vol. 29, pp. 739–748, 2018.

Received: Dec. 16, 2024. Accepted: June 24, 2025