A Fuzzy ELECTRE III Method for Mitigating Malware Attacks on Mobile Devices

Samuel B. Oyong
Department of Computing, Topfaith University, Mkpatak, Nigeria

Uyinomen O. Ekong
Department of Cybersecurity, Faculty of Computing, University of Uyo, Uyo, Nigeria

Victor E. Ekong
Department of Software Engineering, Faculty of Computing, University of Uyo, Uyo, Nigeria

ABSTRACT
Mobile devices are frequently attacked by malware to steal data, credit card information and disrupt operations. The objective of this paper is to develop and hybridize Network intrusion detection system (NIDS) and automated Network intrusion response system (ANIRS) to not only detect malware but provide countermeasures to the attacks and safeguard the target objects. National science laboratory- knowledge discovery in databases (NSL-KDD) dataset was used, although imbalanced. Categorical features such as protocols, services and flags, were converted to numerical values using OneHotEncoder. The dataset was then normalized using min-max normalization technique. Principal component analysis was used to collapse original features to a new but smaller dataset; and then split into training dataset (80%) and test dataset (20%). To develop the NIDS, AdaBoost algorithm was used to train base classifiers such as decision tree, support vector machine and logistic regression, but k-nearest neighbor applied Euclidean distance measure to compute the labels of the target objects. The predictions of the trained models and k-nearest neighbor were aggregated to produce a consensus model called hard vote, which predicts the test dataset to normal and malware labels. Malware labels are grouped into denial of service (DOS), Probe, user to root (U2R) and remote to local (R2L) and passed to ANIRS as input. ANIRS then dynamically generates a set of rules that were used to analyze, prioritize and select optimum response to each attack type. Fuzzy ELECTRE III method was used in prioritizing response actions. Given the imbalanced dataset, the F1-score performance metric of three models that performed best are KNN (91.36%), Hard vote (91.36%) and LR (86.22%). Similarly, countermeasures to malware attacks include DOS: reset connection; Probe: block attacker's IP address; and U2R and R2L: disable user. This paper successfully prioritized countermeasures to detected intrusions and safeguard the target objects.

General Terms
Fuzzy sets, membership function, Jupyter notebook

Keywords
Fuzzy ELECTRE method, Intrusion Detection System, Automated Intrusion Response System, Malware Attack, Optimum Response Selection.

1. INTRODUCTION
The use of mobile devices has become common in homes, industries, schools and government offices as they are used for many purposes: for business, education, commerce, communication and social media. Their diversity, portability and convenience have enticed many users, including malware developers who use them for criminal activities. Malware compromises systems (including mobile devices) and use them to steal information, personal data, blackmail innocent users to ransom, intercept business transactions, and even send and receive messages while the innocent user pays the bills ignorantly [1, 2, 3]. Other motives behind malware attacks include espionage, ideology (terrorism) and fun [4, 5].

The nefarious activities of malware are affecting all sectors of economy. To governments, espionage between nations is on the rise; middle men have been defrauding banks and individuals of their legitimate transactions by stealing their credit/debit card numbers and trade for money [6]; to users, fake application programs are increasing by the day in Google play store, through the use of code injection and third party markets [4]. For instance, online identity theft amounted to 128 cases and non-delivery fraud recorded 65,116 cases in 2018 [6]. Malware damages to systems are on the increase through proliferation of variants of known malware families and resistant to detection by recognizing sand boxes and refusing to launch their payloads while in the controlled environment [2].

To avert these threats, network Intrusion detection systems (NIDS) were hybridized with automated Network intrusion response system (A NIRS). While NIDS monitors the flow of network packets or NetFlow and determines attack traffic from normal traffic at targeted objects or end-points, ANIRS dynamically updates the ruleset for the monitored and transmitted netFlows, and determines the optimum responses to take and counter the attacks [7]. The ruleset is updated based on a set of labeled connections, which should include both normal and abnormal traffic.

A network ruleset is defined as a map r: $X \rightarrow \{0, 1\}$ that takes as input a connection $x \in X$, where X is the set of possible packets or netflows; and outputs a decision $b = r(x) \in \{0, 1\}$. This can be either zero (0) to accept the monitored packet or one (1) to block or redirect the packet from the connection or target.

Labeled connections can be defined as turples:

X = (t, scr_ip, scr_port, dst_ip, dst_port, protocols, …y)

Where:

t: is the first packet's timestamp for the netFlow

src_ip (src_port) are source ip address and port.

dst_ip (dst_port) are destination ip address and port.

Protocol: is the set of rules used by the source and destination hosts to communicate [7].

ANIRS is formally defined as a system that maps

$(r_t, X_A, X_N) \rightarrow r_{t + \Delta t} = R(r_t, X_A, X_N)$

With input as:

r_t: the current network ruleset

X_A: A set of network connections {$x_{A,1}$, $x_{A,2}$, …,} that were labeled as intrusion alerts by NIDS.

X_N: A set of network connections {$x_{N,1}$, $x_{N,2}$, …,} that were labeled as normal traffic by NIDS.

And output as:

$r_{t+\Delta t}$: An updated network ruleset. $\Delta_t > 0$ is the elapsed time during the enforcement of the ruleset [7]

2. LITERATURE REVIEW

Traditionally, network Administrator selects response actions manually based on the temperament, company policy, and response selection criteria [8], [9]. However, given the volume of alerts from distributed NIDSs, it becomes wearisome for system Administrators to cope with and select response actions without delay. Hence the need for an automated network intrusion response system (ANIRS) that will respond to reported attacks in real time, and at reduced cost; and give NIDS the much-needed improved performance [10], [11]]. To overcome these lapses, a number of research contributions have been made.

Multiple Criteria Decision Making (MCDM) process is a decision support approach used in solving cyber security issues of NIDS, which has many attributes and alternative solutions [12]. MCDM has many methods such as analytical hierarchy process (AHP), Elimination Et Choix Traduisant La REalite (translated to Elimination and Choice Expressing Reality) (ELECTRE); Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), preference ranking organization method for enrichment evaluation (PROMETHEE), and so on. In this research work, fuzzy ELECTRE III method is used to select prioritized optimum actions from a set of countermeasures to attacks in real time and at acceptable cost. ELECTRE III has the following advantages over other MCDM types [13]:

i. ELECTRE III method is relatively faster than other methods like AHP, TOPSIS, and PROMETHEE.
ii. ELECTRE III method provides a more realistic decision-making process, and also considers criteria importance weights
iii. Decision makers (DMs) can independently include their own preferences in the decision-making process.
iv. ELECTRE III method is used in solving discrete problems such as classification, ranking, and more.

Security experts, who were interviewed, determined the response alternatives per attack type based on four main security criteria: confidentiality (C), integrity (I), availability (A), and response cost (RC). Criteria are security attributes that attackers attempt to undermine. Given the ambiguity in decision making of domain experts, triangular fuzzy numbers (TFN) are used to translate subjective statements to crisp values. They are mathematical representation of positive and negative assessments of criteria by security experts (DMs) in literature. TFN is a triplet (a, b, c) where;

a = the smallest possible value
b = the most probable value
c = the largest possible value

The membership function $\mathcal{M}$ of TFN is defined as Equation 1:

$$\mathcal{M}_A(x) = \begin{cases} 0, & \text{if } x < a \\ (x-a)/(b-a), & \text{if } a \le x \le b \\ (c-x)/(c-b), & \text{if } b \le x \le c \\ 0, & \text{if } x > c \end{cases} \quad (1)$$

Where x is the horizontal coordinate of a linear scale [14], and the subscript A is the response alternative.

Julia/REPL kernel is used to execute the Fuzzy ELECTRE III method in Jupyter Notebook platform [15].

In [16] a fuzzy ELECTRE 1 method of MCDM was adopted to evaluate consumer preferences in the purchase of smartphones. However, the population size of the survey, being only 250 participants, is considered small. In an attempt to expand a company's operational base, [17] developed a hybrid of AHP and ELECTRE III method to rank and prioritize the subsidiary companies. The work used seven criteria to rank the companies, which were a bit too many for effective operation of ELECTRE methods. The maximum number of criteria effectively used by an ELECTRE method has been five with a large number of alternatives [18]. Also, AHP computes criteria weights objectively, a method more complex than subjective computation and does not consider the opinions of domain experts. In [19], the cost of damage on the target object by intruding applications are compared with that caused by intrusion response actions against the attack. However, the work was semi-automated as it relied on human intervention.

Similarly, [20] attempted to improve the selection process by extending the ELECTRE methods using intuitionistic fuzzy sets (IFS) to handle uncertainties in expert judgments. IFS, also called Type-2 Fuzzy ELECTRE method, used objective computations, ignoring domain experts' opinions. The technique is complex and difficult to compute, resulting in high overhead cost. Another limitation to this work is that the sample population is minimal, only 49 graduate students and 124 college students respectively were considered [[21], [22], [23]].

Similarly, artificial intelligence (AI) and machine learning (ML) automatically analyze massive datasets, detect anomalies, and execute countermeasures with speed and precision. While automation enhances resilience, it also introduces ethical challenges surrounding accountability, fairness, transparency and proportional response [[24], [25], [26]]. Therefore, to balance efficiency with ethical responsibility, automated systems should operate within acceptable moral and legal boundaries. Also, ANIRS should be developed in such a way that it works in tandem with human experts or stakeholders, giving them the results or data needed to make meaningful decisions [27].

Response actions are many and varied. For instance, host-level response provides the following intervening actions:

i. Patch deployment
ii. Service restart
iii. Process isolation, and
iv. Migration

These approaches offer rich recovery operations, but they require privileged access [7], [28] proposed a reinforcement learning (RL) based on IRS that can respond to multi-stage advanced persistent threat (APT) attacks on cyber security. RL maps situations to actions that maximize environmental reward A hybrid AI-SDN (software defined networking) framework for adaptive zero trust security with real-time intrusion response was proposed in [28, 29]. The hybrid defined AI-SDN used AI-driven dynamic trust evaluation, anomaly detection, and

predictive analysis. Indeed, the paper emphasized automated policy adaptation, proactive containment, and distributed enforcement to minimize attack impact.

In [30], a survey was carried out on AI-based responses to cyber threats. The study aimed at hybridizing cybersecurity with AI, which goes beyond IRS to include algorithms, frameworks and architectures. In another development, In [31], the reasons why research on IRS in literature was very slow compared to that of IDS were highlighted to include:

i. The pool of available counter measures to attacks is constrained.
ii. Limited knowledge exists in the combination of multiple atomic responses when countering identified attacks.
iii. Scarcity of accessible datasets for the construction of offline based response systems is largely undermined.
iv. Open-source tools that provide prevention, evaluation, and response are lacking, resulting in a shift of interest by researchers from IRS. Even more, the possibility of providing a more costly response compared to attack cost, cannot be over ruled.

To curb the menace of cyber threats, deep learning neural network (DLNN) algorithms are usually used. However, DLNN methods are shrouded in secrecy (black box) and results generated from their use are not popular in critical settings like healthcare, finance, the military, and more [32]. However, explainable AI is being integrated to these uses to reverse the trend.

The objective of this research work is to solve the problem of malware intrusions by providing detection and control measures, automating the process and using a dataset of 125,973 training records with 22,544 test data records. Ensemble learning boosting technique is used to train models that developed NIDS, and Julia/REPL was used to effect the automation of NIRS in Jupyter notebook platform.

The remaining part of this paper is subdivided into the following subsections: Section 2 discusses the methods used to develop NIDS and prioritize response alternatives in relation to the attack types using ANIRS. While section 3 presents the results obtained in the analysis and compares the results obtained with that of other works in literature. Section 4 concludes the paper and made suggestions for further works.

3. METHODOLOGY

3.1 Datasets

The dataset used in this work is national science laboratory-knowledge discovery in databases (NSL-KDD) [33]. It is imbalanced and contains categorical features, which were converted to numerical values using oneHotEncoder function before being used to train the models. Security experts rated the features using different units and measurements, which had to be normalized using min-max normalization technique. Dimensionality reduction was carried out using principal component analysis (PCA), which collapsed the features and extracted a much smaller set, but maintained the characteristics of the original features.

To develop the NIDS, first the data was split into training dataset (80%) and test dataset (20%). Then, Adaboost algorithm, an ensemble boosting technique, was used to sequentially train base models after weighting the training features in memory. The trained models predicted the records' labels of the training dataset. K-nearest neighbor does not support the weighting process of Adaboost, and rather applied Euclidean distance measure to compute the k nearest neighbors to the target object, and assigned the label of the set with majority vote to the object. These predictions were aggregated using voting classifier to form a consensus model called hard vote, which was used to classify the test dataset into normal and abnormal (malware) labels. The abnormal labels were categorized into four: DOS, Probe, U2R and R2L. These attack types and the normal types were sent to the automated NIRS, which dynamically assessed the ruleset per attack type to select the optimum response option provided by ELECTRE III method in Julia/REPL for counter action to the respective attack type.

The security criteria undermined by these attack types, for this work, include confidentiality (C), integrity (I), availability (A), and response cost (RC). The response alternatives for each attack type in literature are depicted in Table 1.

Table 1: Response alternatives per attack type.

Attack types	Response Alternatives	
DOS	i.	Shutdown Host (SH)
	ii.	Reboot Host (RH)
	iii.	Restart Process (RP)
	iv.	Reset Connection (RST)
PROBE	i	No Response (NR)
	ii	Block Attacker's Source IP Address (BSIP)
	iii	Close Destination Port (DP)
U2R and R2L	i	Shutdown Host (SH)
	ii	Kill Process (KP)
	iii	Disable User (DU)

In Table 1, there are three groups of attack types. U2R and R2L are grouped together because they have a common purpose.

3.2 Preprocessing of Data

Steps used to achieve the selection of prioritized response are contained in NIRS architecture, depicted in Figure. 1

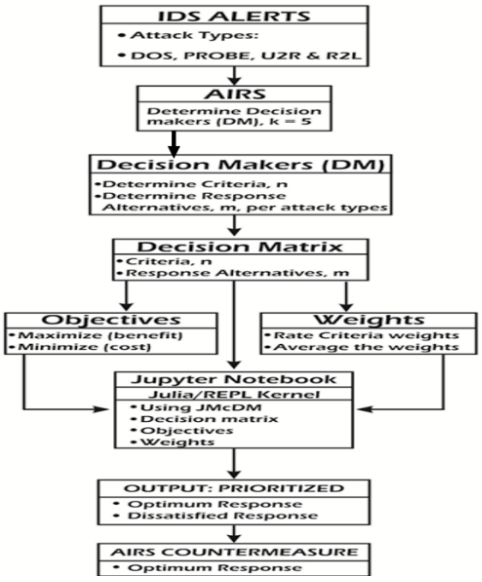

Figure 1: NIRS Architectural design Using Fuzzy ELECTRE Method

From Figure 1, the predictions of malware attack types (DOS, Probe, U2R and R2L) are sent to ANIRS as input values. NIRS then dynamically determined the ruleset for actions based on the

attack type. The response alternatives have been determined in literature, as depicted in Table 1, with m=4, 3, 3 for DOS, Probe, and U2R/R2L respectively. The decision matrix is then constructed using criteria and response alternatives; weights and objective vectors are also determined. While the weights represent the averaged criteria ratings, the objectives represent whether the choice of the response will maximize the resulting action (benefit) or minimize the resulting action (cost). These parameters are then input into Julia/REPL using Jupyter Notebook platform. Julia/REPL then uses ELECTRE function to determine the optimum response, and the worst response respectively, and output to ANIRS, which choses the optimum response and provide the countermeasure, per attack type. The optimum response provides the countermeasure with minimum cost in real time. However, if the dissatisfied response is used, for whatever reason, the response cost would outweigh the attack cost. That would not be desired.

4. RESULTS AND DISCUSSIONS

4.1 Results

Table2 presents the predictions of hard vote classifier or model using test dataset. For want of space, we have presented the first eleven records, first seven features or columns and the last seven features containing expected (actual) and predicted labels

Table 2: Hard Vote predictions and test datasets

	duration	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot
0	5	2429	475	0	0	0	0
1	0	45	134	0	0	0	0
2	0	45	80	0	0	0	0
3	1979	145	105	0	0	0	0
4	14462	1	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	294	5499	0	0	0	0
7	0	0	0	0	0	0	0
8	0	1032	0	0	0	0	0
9	0	1	0	0	0	0	0
10	0	0	0	0	0	0	0

flag_S2	flag_S3	flag_SF	flag_SH	model_prediction_number	actual_value	model_prediction
0	0	1	0	1	Normal	Normal
0	0	1	0	1	Normal	Normal
0	0	1	0	1	Normal	Normal
0	0	1	0	1	Normal	Normal
0	0	0	0	2	Probe	Probe
0	0	0	0	0	DoS	DoS
0	0	1	0	1	Normal	Normal
0	0	0	0	0	DoS	DoS
0	0	1	0	0	DoS	DoS
0	0	1	0	2	Probe	Probe
0	0	0	0	0	DoS	DoS

Supervised learning is adopted where a model is trained to predict the labels of the test dataset and compare with the actual (known) labels. The test dataset acts as surrogate to real-world dataset that the trained model will be working with to test its generality and performance. Confusion matrix was used to compare the predicted labels with the actual labels, and recorded true positive (TP), true negative (TN), false positive (FP) and false negative (FN) predictions. Figure 2 depicts confusion matrix of hard vote classifier, with TP lying along the main diagonal in a multiclass confusion matrix.

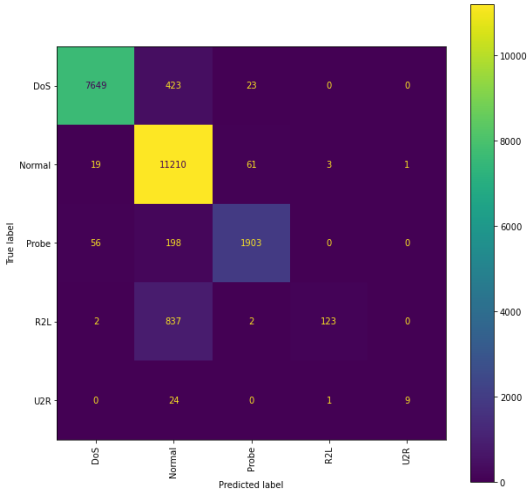

Figure 2: Confusion Matrix of Hard vote classifier

However, the models were evaluated for performance and results were presented in Table 2. Because the dataset was imbalanced, accuracy values became unreliable as it measures overall correctness without distinguishing between the class labels, using values of majority classes and ignoring that of minority classes, which could be misleading (providing accuracy paradox). Since Recall measures the ability to find all minority class instances, and Precision measures the accuracy of the positive predictions. The onus now falls on F1 score, which is the harmonic mean of both Recall and Precision and will offer a balance. Therefore, F1-score values were used to showcase the best performing model with scores: 91.36% (hard vote) as expected, KNN has good result as that of hard vote model, as depicted in Table 2.

Table 2: Performance evaluation of the models

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
AdaBoost + SVM	85.14	84.96	85.14	83.31
KNN	92.69	93.23	92.69	91.36
AdaBoost + Logistic Regression	87.74	86.73	87.74	86.22
AdaBoost + Decision Tree	85.18	85.41	85.18	84.22
Voting Classifier	92.68	93.23	92.68	91.36

SVM and DT did not do well in this analysis, partly because of the preprocessing techniques applied. More so, SVM is not efficient in multiclass classification but binary classification, and there is the problem of kernel selection in the analysis. It is also not good in linear space classification, but high dimensional spaces; similarly, DT is sensitive to slight variations in data, resulting in high variance and over fitting. DT can also struggle with complex relationships and noisy data [34].

For the control process, NAIRS prioritized the response actions, following the generated ruleset and results from Julia/REPL kernel. Julia/REPL kernel process is as follows:

4.1.1 DOS attack Type

The parameters used include;

Criteria, n = 4, [C, I, A, RC]
Response Alt., The command cell snippet is displayed as In[23] and Out[23], where In[ ] is input and Out[ ] is output respectively. The number inside the square braces are auto filled represents the kernel order of execution. And df1 is the dataFrame representing the decision matrix. A screenshot of the result using Julia/REPL for DOS attack type is depicted in Figure 3. REPL is an interactive command line interface, which stands for Read-Evaluate-Pring-Loop. It allows users to evaluate Julia code in real time.

m = 4, extracted from Table 1.

1 →SH, (Shutdown Host)
2 → RH, (Reboot Host)
3 → RP, (Restart Process)
4 →RST, (Reset Connection)

The command cell snippet is displayed as In[23] and Out[23], where In[ ] is input and Out[ ] is output respectively. The number inside the square braces are auto filled and represents the kernel order of execution. df1 is the dataFrame representing the decision matrix. A screenshot of the result using Julia/REPL for DOS attack type is depicted in Figure 3. REPL is an interactive command line interface, which stands for Read-Evaluate-Pring-Loop. It allows users to evaluate Julia code in real time.

As mentioned earlier, the dataset is imbalanced. This led to the fact that R2L and U2R virtually had little or no values to predict and compare. The normal records had 11,210 true positive (TP) predictions, followed by DOS with 7,649 TP records, Probe had 1,903 TP records. R2L attack type had very small records 123, and U2R had no records to predict an compare, recording zero (0) in confusion matrix, as depicted in Figure 3.

```
In [23]: using JMcDM
         df1 = DataFrame(
         C = [9.67, 7.00, 5.00, 3.00],
         I = [9.67, 7.00, 5.00, 3.00],
         A = [0.33, 3.00, 7.00, 8.67],
         RC = [9.67, 8.67, 5.00, 3.00]);
         weights = [0.20, 0.17, 0.49, 0.14];
         objectives = [max, max, max, min];
         result = electre(df1, weights, objectives);
         result.bestIndex

Out[23]: (4, 1)

In [24]: objectives

Out[24]: 4-element Vector{Function}:
          max (generic function with 27 methods)
          max (generic function with 27 methods)
          max (generic function with 27 methods)
          min (generic function with 27 methods)

In [25]: weights

Out[25]: 4-element Vector{Float64}:
          0.2
          0.17
          0.49
          0.14

In [26]: df1

Out[26]: 4 rows × 4 columns
```

	C	I	A	RC
	Float64	Float64	Float64	Float64
1	9.67	9.67	0.33	9.67
2	7.0	7.0	3.0	8.67
3	5.0	5.0	7.0	5.0
4	3.0	3.0	8.67	3.0

Figure 3: Screenshot of Julia/REPL Results and Hyper parameters for DOS attack type

Julia/REPL kernel is used to run the code in Jupyter Notebook platform [16]. From Figure 3, the code snippet is keyed into In[23] input cell. The corresponding out[23] is the output of the computed result that is sent back to the cell for display. Out of the four (4) response alternatives, the preferred response selected is 4, which is ResetConnection, as depicted in Figure1, and will incur minimum cost. The second number in Out[23] indicates that it is the least preferred option and will incur cost that will be higher than the damage done by the attack type. This will not be desirable.

Accessing the Hyper Parameters of the Code Snippet

The essence of accessing the hyper parameters of the code snippet is to offer explanation on their roles.

From Figure 3, Objectives depict effects of criteria on response alternatives, whether a particular response alternative will be beneficial (maximize effect) or detrimental (minimize effect). For instance, the first three criteria are beneficial to the response action, while response cost (RC) reduces the effect of the response action.

Also, from Figure 3, the weights depict the view of each DM on the criterion with respect to each response alternative. More so, the criteria are not of equal importance [11]. The sum of all the weights should be equal to 1. They are of type float with 64 bit word length.

Similarly, the decision matrix, represented in Pandas dataframe (df1), consists of the four rows of response alternatives, and four columns of criteria used to determine the optimum response. The

values are Float64 data type, as depicted in Figure 3. Although the rows are numbered, they represent SH, RH, RP, and RST respectively.

4.1.2 Result for PROBE Attack Type

The parameters used include:

Criteria, n = 4, [C, I, A, RC]

Response Alt. m = 3; [NR, BSIP, CDP]

Where;

1: NR → No Response

2 BSIP → Block Attackers' Source IP Address

3 CDP → Close Destination Port

The command cell snippet is displayed as In[3] and Out[3], where In[ ] is input and Out[ ] is output respectively. The number inside the square braces represents the kernel order of execution. df2 is the dataframe representing the decision matrix for PROBE attack type, as depicted in Figure 4.

```
In [3]: using JMcDM
        df2 = DataFrame(
            C = [ 0.33, 7.00, 8.67],
            I = [0.33, 7.00, 8.67],
            A = [0.33, 8.67, 3.00],
            RC = [0.33, 0.33, 7.00]);
        weights = [0.15, 0.15, 0.36, 0.34];
        objectives = [maximum, maximum, maximum, minimum];
        result = electre(df2, weights, objectives);
        result.bestIndex

Out[3]: (2, 1)

In [4]: df2

Out[4]: 3 rows × 4 columns

                C        I        A        RC
            Float64  Float64  Float64  Float64
        1    0.33     0.33     0.33     0.33
        2    7.0      7.0      8.67     0.33
        3    8.67     8.67     3.0      7.0

In [5]: weights

Out[5]: 4-element Vector{Float64}:
         0.15
         0.15
         0.36
         0.34

In [6]: objectives

Out[6]: 4-element Vector{Function}:
         maximum (generic function with 16 methods)
         maximum (generic function with 16 methods)
         maximum (generic function with 16 methods)
         minimum (generic function with 17 methods)

In [7]: print(result)

        Best indices:
        (2, 1)
```

Figure 4: Screenshot of Julia/REPL Result and hyper parameters for PROBE attack type

From Figure 4, df2 contains the criteria [C, I, A, and RC] as rows, but the response alternatives are represented as columns. However, when the hyper parameter df2 is assessed, the Table is properly arranged. Out[3] result displayed is alternative response 2, which is block attacker's source IP address (BSIP). BSIP is the preferred response over the other two. In consequence, ANIRS will select this option to avert the attack type. The second number, option 1, in Out[3] represents the least preferred option, as it will incur response cost which may be even higher than the damage done by the attacker.

4.1.3 Result for U2R and R2L Attack Types

The parameters used include:
Criteria, n = 4
Objective Vector = [max, max, max, min]
Response Alt., m = 3, they are:
1: SH – Shutdown Host
2: KP – Kill Process
3: DU – Disable User

The numbered response alternatives are three. Julia/REPL kernel in Jupyter Notebook platform is used to prioritize the response options. The decision matrix df3, weights and objectives are keyed into the command cell, In[11], starting with "using JMcDM", as depicted in the screenshot of results in Figure 5.

```
In [11]: using JMcDM
         df3 = DataFrame(
             C = [8.67, 8.67, 8.67],
             I = [8.67, 8.67, 8.67],
             A = [0.33, 5.00, 8.67],
             RC = [8.67, 5.00, 0.33]);
         weights = [1.00, 1.00, 1.00, 1.00];
         objectives = [max, max, max, min];
         result = electre(df3, weights, objectives);
         result.bestIndex

Out[11]: (3, 1)

In [7]: df3

Out[7]: 3 rows × 4 columns

         |  | C | I | A | RC |
         |---|---|---|---|---|
         |  | Float64 | Float64 | Float64 | Float64 |
         | 1 | 8.67 | 8.67 | 0.33 | 8.67 |
         | 2 | 8.67 | 8.67 | 5.0 | 5.0 |
         | 3 | 8.67 | 8.67 | 8.67 | 0.33 |

In [8]: print(result)

         Best indices:
         (3, 1)

In [9]: weights

Out[9]: 4-element Vector{Float64}:
         1.0
         1.0
         1.0
         1.0

In [10]: objectives

Out[10]: 4-element Vector{Function}:
         max (generic function with 27 methods)
         max (generic function with 27 methods)
         max (generic function with 27 methods)
         min (generic function with 27 methods)
```

Figure 5: Screenshot of Julia/REPL Result with hyper parameters for U2R and R2L attack types

From Figure 5, it is observed that the optimum option preferred is displayed in Out[11] output as 3, which is Disable User (DU). Along with the optimum option, is the least preferred option 1, which is shutdown host (SH).

4.2 Discussions

Malware is software that perpetrates evil. Malware steals data, disrupts operations, takes over the function of operating system and enables backdoor entry of other evil variants [[36], [35]]. The objective of this paper is to detect the presence of malware in the network using NIDS so as to halt its progression through ANIRS in real time. The Inference engine (IE), represented by ANIRS in this paper, used Fuzzy ELECTRE III method to prioritize and selects optimum response per attack type to be:

i. For DOS attack type, Reset Connection (RST)

ii. For Probe attack type, Block Attacker's Source IP Address (BSIP)

iii. For U2R and R2L attack types, Disable User (DU)

However, [9] combined Fuzzy AHP, Fuzzy TOPSIS and Fuzzy Rule Based Inference Engine to select response action per attack type to be:

i. For DOS attack type, RST was prioritized and selected

ii. For Probe attack type, BSIP was prioritized and selected

iii. However, for U2R and R2L attack types, Shutdown Host (SH) was prioritized and selected as the optimum response for the attack. This choice differed from the DU prioritized and selected in this paper.

The work in [9] used confidence index and target object value as criteria. These parameters were not considered in this paper, rather confidentiality, integrity, availability and response cost were considered. The paper in [9] objectively combined Fuzzy AHP, Fuzzy TOPSIS and Fuzzy rule base to achieve its results. However, this paper combined Fuzzy ELECTRE III method, subjective experts' judgments, and triangular fuzzy numbers (TFN) to convert the vague judgments to crisp values for analysis [20] categorized normalization techniques and recommended the use of linear-sum normalization for a particular MCDM method, simple adaptive weighting (SAW) method. However, this paper went a step further to use two normalization techniques; linear-sum and vector normalization to improve the quality of the dataset for the selected learning algorithms.

Furthermore, [20] used intuitionistic fuzzy sets (IFS) to handle uncertainty in decision making judgments, and objectively outranked customer decisions using score function. IFS is an improvement on fuzzy set (FS) theory, but it is complex, difficult to compute with high overhead, and security experts tend to despise the objective computations because of ethical issues involved as a result of their non-involvement [18]. Given this reality, the paper rather adopted interval valued fuzzy sets (IVFS) to compute uncertainty, which is simpler than IFS.

However, in this paper, the opinions of five security experts were sought for in ranking the criteria and providing response alternatives for the attack types. It used simple FS theory to subjectively convert confusable, imprecise and vague judgments of decision makers, and combined this process with ELECTRE III method to select optimum response to malware attack, per attack type, in real time, devoid of human intervention.

Again, [20] had used a minimal population size of 49 graduate students and 124 college students in gathering data for the

market survey of preferred mobile devices. On the contrary, this paper used 125,973 training records, and 22,544 test records for training and generalization. More so, this paper employed 67,343 normal records to train the base classifiers of the behavior and characteristics of normal applications [37], which used that knowledge to detect malware, without reference to family signatures of malware types stored in databases.

5. CONCLUSION

Malware intrusion on mobile devises has not only been detected but controlled. The detection aspect was carried out through the design and implementation of NIDS using boosting technique. As a backend process, NIDS alerts were reported to ANIRS. ANIRS used Fuzzy ELECTRE III method to prioritize and select optimum response per attack type. The use of Fuzzy ELECTRE III method to prioritize, select and foil attack by ANIRS in real time and at minimal cost has enabled the following achievements and contributions to knowledge:

i. NIDS performance has been enhanced by hybridizing with ANIRS

ii. The combination of NIDS and ANIRS to prioritize, select and foil reported attacks, using Fuzzy ELECTRE III method was the first of its kind, to the best of our knowledge (a novelty).

iii. The ANIRS functioned without human intervention, which saved time and protected target objects at minimal cost.

iv. Criteria weighting, which used to be manually assigned by system administrators, has now been automated.

However, security attributes are many. In this paper, only four are used: Confidentiality, integrity, availability and response cost. Other security criteria such as IDS confidence index, time/date, target resource value, command, and attack agent should be used to prioritize, select and foil attack using Fuzzy ELECTRE III method, and compare results.

6. ACKNOWLEDGMENTS

The authors thank the network security experts consulted in the course of the research.

7. REFERENCES

[1] Gamao, A. O. 2018 Malware Analysis on Android Applications: A Permission Based Approach. *ResearchGate.* DOI: 10.18535/SSHJ/v2i10.109.

[2] Inayat, Z., Garis, A., Anuar, N. B; Khan, M. K. 2016, Intrusion Response Systems: Foundations, Design, and Challenges. *Journal of Network and Computer Applications, 62, 53 – 74, http://dex.doi.org/10.1016/j.jnca.2015.006*

[3] Atkinson, M. 2015, An Analysis of Android Application Permissions. *Internet and Technology.* Pew Research Center.

[4] Verizon 2018. *Data Breach Investigations Report*, www.verizonenterprise.com/Federal.

[5] Group Special Mobile Association (GSMA) 2019. *Mobile telecommunications security threat landscape*. Floor 2, the Walbrook Building, 25 Walbrook, London EC4N 8AF United Kingdom.

[6] Stein, J. 2020. *Data Breach Report*, North Carolina Department of Justice. *www.ncdoj.gov/complaint*

[7] Marchioro,T., Saroui, R., Oliverau, A, 2025. Network Intrusion Response System: Towards Standardized Evaluation of Intrusion Response. *ESORICS: 30TH European Symposium of Research in* Toulouse, France. Hal- 05294762. https://hal.scioence/hal-05294762

[8] Singh, D.K. & Kaushik, P. 2018. Framework for Fuzzy Rule Base Automatic

intrusion Response Selection System (FRA IRSS) Using Fuzzy AnalyticHierarchy Process and Fuzzy TOPSIS. *Journal of Intelligent and Systems,* DOI: 10.3233/JIFS- 18350

[9] Shojaie, A. A., Babaie, S., Sayah, E., & Mohammeditabar, D. 2016. Analysis and Prioritization of Green Health Suppliers Using Fuzzy ELECTRE Method with a Case Study. *Global Journal of Flexible Systems Management.* DOI: 10.1007/s40171- 0171-017-0168-2

[10] Singh, D.K. & Kaushik, P. 2016. Analysis of Decision Making Factors for Automated Intrusion Response System (AIRS): A R*eview*. *International Journal of Computer Science and Information Security (IJCSIS),* 14(6)

[11] Singh, D.K. & Kaushik, P. 2019. Intrusion Response Prioritization Based on Fuzzy ELECTRE Multiple Criteria Decision-Making Technique. *Journal of Information Security and Applications,* 48, 102359.

[12] Alamleh, A., Albahri, O. S., Zaidan, A. A., Alamoodi, A. H., Albahri, A. H., Albahri, A. S., Zaidan, B. B., Qahtan, S., Binti Ismail, A.R., Malik, R. O., Baqer, M. J., Jasin, A. N., Al-Samarraay, M.S. 2022. Multi-Attribute Decision Making for Intrusion Detection Systems: A Systematic Review. *International Journal of Information Technology and Decision Making,* 22(01), 589 – 636, DOI: 10.1142/S021962202230004X

[13] Horta, A. 2021. Scikit-MCDM: The Python Library for Multi-Criteria Decision aid, ver. 0.21 [Open source. Available in https://github,com/cybercrafter.com

[14] Komsiyah, S., Wongso, R., Scalisi, S.W. 2019. Applications of the fuzzy ELECTRE method for decision support systems of cement vendor selection. 4th International Conference on Computer Science and Computational Intelligence (ICCSCI), 157, 479 – 488.

[15] Satman, M. H., Yildirim, B.F. & Kuruca, E. 2021. JMcDM: A Julia Package for Multiple Criteria Decision–Making Tools. *The Journal of Open Source Software*.

[16] Belbag, S., Gungordu, A., Yumusak, T., and Yilmaz, K. G. 2016. The Evaluation of Smartphone Brand Choice: Application With the Fuzzy ELECTRE I Method. *International Journal of Business and Management Invention,* 5(3), 5 – 63.

[17] Yucel, M.G. & Gorener, D. 2016 Decision making for company acquisition by ELECTRE method. *International Journal of Supply Chain Management,* 5(1).

[18] Odu, G. O., 2019. Weighting Methods or Multi-criteria Decision Making Techniques. *Journal of Applied Science Environmental Management,* 23(8): 1449 - 1457. https://dx.doi.org/10.4314/jasen.v23i87

[[19] Nejat, S.K. & Kabiri, P. 2017. An adaptive and cost-based intrusion response system, Cybernetic *and Systems*. DOI: 10.1018/01969722.2017.1319693.

[20] Wu, M. 2019. Comparative study of ELECTRE methods with intuitionistic fuzzy sets applied on consumer decision making case. EJERS, *European Journal of Engineering Research and Science,* 4(10).

[21] Akmaludin, B. M., Marlinda, L., Dalix, S.S., & Santoso, B. 2018. *The Employee Promotion Based on Specification job's Performance Using: MCDM, AHP and ELECTRE METHODS.* The 6th International Conference on Cyber and IT Service Management (CITSM).

[22] Khan, M. & Ansari, M. D. 2020. Multi-criteria software quality model selection based on divergence measure and score function. *Journal of Intelligent and Fuzzy Systems* 38(2020) 3179-31-3188. DOI:10.3233/JIFS-191/153.

[23] Liu, Y. & Du, J. A. 2020. multi-criteria decision support framework for renewable energy storage technology selection. *Journal of Cleaner Production* 277 (2020), 122183. https://doi.org/10.1016/j.jclepro.2020.122.183

[24] Sonkar, N. 2025. Establishing a Cybersecurity and Privacy Practice in Mid-Sized Consulting. *A Blueprint for Scalable Risk Adversary Services.* https://dx.doi.org/10.2139/ssrn.5253056

[25] Panchal, P. B. 2025. Use of Integrated Intelligence Scheduling System (IISS for Heavy Civil Construction Projects. *Journal of Emerging Technology and Innovative Research, 12(5):a800 – a815. https://doi.org/10.56975/jetir.v12i5.560867*

[26] Singh, B. 2025. Securing the Network Future: AI, Resilience and Human Centric Design. *Notion Press, https://direct.notionpress.com/in/read/securing-the-ntelligence-future-and-human-centric-design-hardcover/*

[27] Kumar, L.K.S., Uyyala, R., Gera, J., Asulu, A.L.S., Sasirekha, P., Krishna, K.R. 2025. AI-Powered Intrusion Response for Intelligent Vehicular Ecosystem. *Journal of Theoretical and Applied Information Technology, 103(16):6279-6288.*

[28] Iturbe, E., Rego, A., Llorete-Vaquez, O., Rios, E., Dalamagkas, C., Merkouris, D., Toledo, N. 2025. Reinforcement Learning in Actin: Powering Intelligent Intrusion Responses to Advanced Cyberthreats in Realistic Scenarios. *Expert Systems with Applications, 296(2026)129168. https://doi.org/10.1016/j.eswa.2025.129168*

[29] Tiger, Z. and Smith, J. 2025. Hybrid AI-SDN Framework for Adaptive Zero Trust Security with Real-Time Intrusion Response. *Multidisciplinary Innovations and Research Analysis, 6(1):21 – 27.*

[30] Molina, S.B., Marmol, F.C., Nespoli, P. 2024. Tackling Cyber Attacks Through AI-Based Reactive Systems: A Holistic Review and Future Vision. arXiv:2312.06229v2 [cs.CR].

[31] Bashendy, M., Tantawy, A., and Erradi, A. 2024. Autonomous Response Agent for Cyber Physical System Attacks: A Model-Free Deep Reinforcement Learning Approach (DRL-IRS). ResearchGate. https://dx.doi.org/10.2139/ssrn.4716080

[32] Xue, Q., Zhang, Z., and Wang, M. 2025. Industrial Internet Response Based on Explainable Deep Learning. *Electrnics,* 14, 987.

[33] Bala, R. and Nagpal, R. 2019. A review on kdd cup99 and nsl-kdd dataset. *International Journal of Advanced Research in Computer Science*, *10*(2).

[34] Arifuddin A., Buana G.S., Vinarti R.A., Djunaidy A. 2024. Performance comparison of decision tree and support vector machine algorithms for heart failure prediction. *Procedia Computer Science*, 1234:628-36.

[35] Belal, M. M. and Sundaram, D. M. 2023. An Intelligent Protection Framework for Intrusion Detection in Cloud Environment Based on Covariance Matrix Self-adaptation Evolution Strategy and Multi-Criteria Decision Making. *Journal of Intelligent and Fuzzy Systems,* 44(6), 8971 – 9001. DOI: 10.3233/JIFS-224135

[36] Madhavi, S., Santhosh, N, C., Rajkumar, S., and Praveen, R. 2023. Pythagorean Fuzzy Set Based VIKOR and TOPSIS Based Multi Criteria Decision Making Modelsor Mitigating Resources Deletion Attacks in WSNs. *Journal of Intelligent and Fuzzy Systems,* 44(6), 9441 – 9459. DOI: 10.3233/JIFS-224141

[37] GitHub Inc. 2020. NSL-KDD Dataset, https://www.github.com