Research Article

A Novel Approach for Classifying MANETs Attacks with a Neutrosophic Intelligent System based on Genetic Algorithm

Haitham Elwahsh [1], Mona Gamal,[2] A. A. Salama,[3] and I. M. El-Henawy[4]

[1]*Computer Science Department, Faculty of Computers and Information, Kafrelsheikh University, Kafrelsheikh 33516, Egypt*
[2]*Information System Department Faculty of Computers and Information, Kafrelsheikh University, Kafrelsheikh 33516, Egypt*
[3]*Department of Mathematics and Computer Science, Faculty of Sciences, Port Said University, Port Said 522, Egypt*
[4]*Computer Science Department, Faculty of Computers and Information, Zagazig University, Zagazig, Egypt*

Correspondence should be addressed to Haitham Elwahsh; haitham.elwahsh@gmail.com

Recently designing an effective intrusion detection systems (IDS) within Mobile Ad Hoc Networks Security (MANETs) becomes a requirement because of the amount of indeterminacy and doubt exist in that environment. Neutrosophic system is a discipline that makes a mathematical formulation for the indeterminacy found in such complex situations. Neutrosophic rules compute with symbols instead of numeric values making a good base for symbolic reasoning. These symbols should be carefully designed as they form the propositions base for the neutrosophic rules (NR) in the IDS. Each attack is determined by membership, nonmembership, and indeterminacy degrees in neutrosophic system. This research proposes a MANETs attack inference by a hybrid framework of Self-Organized Features Maps (SOFM) and the genetic algorithms (GA). The hybrid utilizes the unsupervised learning capabilities of the SOFM to define the MANETs neutrosophic conditional variables. The neutrosophic variables along with the training data set are fed into the genetic algorithm to find the most fit neutrosophic rule set from a number of initial subattacks according to the fitness function. This method is designed to detect unknown attacks in MANETs. The simulation and experimental results are conducted on the KDD-99 network attacks data available in the UCI machine-learning repository for further processing in knowledge discovery. The experiments cleared the feasibility of the proposed hybrid by an average accuracy of 99.3608 % which is more accurate than other IDS found in literature.

1. Introduction

Mobile Ad hoc Network (MANET) is a group of wireless mobile hosts shaping an impermanent system without communication infrastructure. This network may change faster and unpredictably. The exceptional qualities of MANETs give a foe the chance to dispatch numerous assaults against especially ad hoc networks [1, 2]. Security in MANETs is the most imperative worry for the fundamental usefulness of system. Accessibility of system administrations, privacy, and uprightness of the information can be accomplished by guaranteeing that security issues have been met.

MANETs suffer from the unwell effects of security assaults on accounts, in respect of its characteristics like open medium, changing its topology powerfully, absence of focal checking and administration, agreeable calculations, and no unmistakable safeguard system. These elements have changed the war zone circumstance for the MANETs versus the security menace. These features make MANETs more impotent to be a victim by an assailant from inside the network. Remote connections likewise make the MANETs more vulnerable to assaults which make it simpler for the assailant to enter the system and access the progressing communication [3–5]. Mobile nodes inside the connection area can hear and cooperate in the network.

MANETs must have a protected path for transmission and correspondence; this is very defying and fundamental issue as there are expanding dangers of assault on the Mobile Network. Security is the cry of the day. To give secure transmission and communication, designers must comprehend distinctive kinds of assaults and their impacts on the MANETs. MANET can suffer from various threats such as

routing table over flow, flooding attack, wormhole attack, Sybil attack, denial of service (DoS), black hole attack, and selfish node misbehaving. Nodes in Mobile Ad Hoc Networks (MANETs) are without any predefined infrastructure and mobility; then they are susceptible for intrusion and attack. Therefore, designer's use of Intrusion Detector Learning Software is to detect network intrusions and protect a computer network from unauthorized users, including perhaps insiders. Building a predictive model (i.e., a classifier) using intrusion detector is a learning task. The detector should be capable of distinguishing between "abnormal" connections, called intrusions or attacks, and normal connections. The 1998 DARPA Intrusion Detection Evaluation Program was done and controlled by MIT Lincoln Labs. Intrusion Detection Systems (IDS) suffer from uncertainty and imprecise nature. Neutrosophy theory introduced by Smarandache [6] could be utilized to suit the ambiguity nature of the IDS. In [7], Salama presented the principle of Neutrosophic Set (NS) and mathematical theory, to define any situation by a ternary crisp build. Salama et al. work [6, 8] formulated a beginning to new fields of neutrosophic theory in computer discipline. The neutrosophic indeterminacy assumption is very significant in many of circumstances such as information fusion (collecting data from various sensors). Also, NS is a conceivable common traditional system that generalizes the principle of the traditional set, Fuzzy Set (FS) [9] and Intuitionistic Fuzzy Set (IFS)) [10], etc. NS 'A' determined on universe U. $x = x(\mathbf{T}, \mathbf{I}, \mathbf{F}) \in$ A with $\mathbf{T}, \mathbf{I}$, and $\mathbf{F}$ are defined over the interval $]\mathbf{0}^-, \mathbf{1}^+[$. $\mathbf{T}$ is the truth-MEMEBERSHIP, $\mathbf{I}$ is the INDETERMINACY, and F is the falsity-MEMEBERSHIP degrees on the set $\mathbf{A}$.

Designing a neutrosophic IDS is a proper solution in handling vague circumstances. The neutrosophic IDS is formed of two subphases: the preprocessing stage and the network attacks classification stage. The preprocessing stage is concerned with formulating the network features in a format appropriate for the classification. The KDD network data [11] is reformatted into neutrosophic form $(\mathbf{x}, \boldsymbol{\mu}_\mathbf{A}(\mathbf{x}), \boldsymbol{\sigma}_\mathbf{A}(\mathbf{x}), \boldsymbol{\nu}_\mathbf{A}(\mathbf{x}))$, where x is the value of feature, $\boldsymbol{\mu}_\mathbf{A}(\mathbf{x})$ is the MEMEBERSHIP (MEM), $\boldsymbol{\sigma}_\mathbf{A}(\mathbf{x})$ is the INDETERMINACY (I), and $\boldsymbol{\nu}_\mathbf{A}(\mathbf{x})$ is the NONMEMEBERSHIP (NON_MEM) degrees of the x in the feature space. The Self-Organized Features Maps (SOFMs)[12], machine-learning technique, was used to prepare the neutrosophic KDD through learning the MEM, NON_MEM, and I functions of the KDD network attacks data [11] downloaded from the UCI repository for further processing in knowledge discovery [13]. After converting the traditional KDD data into a neutrosophic one, the genetic algorithms (GAs) [14] searching mechanism is utilized in finding a set of neutrosophic (if-then) rules to classify MANETs attacks. The GA initial population is a set of randomly generated individuals. Each individual represents a structure of a neutrosophic (if-then) classification rule. During the GA iterations, the selection, crossover, and mutation processes are applied on the populations for generating new fit offsprings. The fitness of the offsprings is quantified by the concept of neutrosophic correlation coefficient introduced by Salama et al. in [12]. The final population will serve as the neutrosophic rule set for the

neutrosophic IDS for the KDD data set. The testing procedure applies new instances of KDD instances (not used during training) to measure the accuracy levels of the obtained neutrosophic IDS. The experiments compared the proposed neutrosophic IDS with a number of well-known classifiers in literature like C4.5 [15], SVM [15], ACO [16], PSO [17], and EDADT [18]. The comparisons proved the feasibility of the proposed neutrosophic IDS in terms of accuracy level of average 99.3608 %and false alarm rates of average 0.089 %.

The rest of this paper is organized as follows: Section 2 presents the theories and overviews. Section 3 proposes the GA in generating the neutrosophic (if-then) rules (inference engine). Experimental results and conclusion are shown, respectively, in Sections 4 and 5.

2. Theories and Overview

2.1. Motivation and Related Work. Mobile Ad hoc Network (MANET) is a system of wireless mobile nodes that are dynamically self-organized in arbitrary and temporary network topologies without communication infrastructure. This network may change quickly and unforeseeably. The unique characteristics of MANET give an adversary the opportunity to launch numerous attacks against ad hoc networks [2]. Security in Mobile Ad Hoc Network is the most important concern for the basic functionality of network [5]. The security issues have been met when availability of network services, confidentiality, and integrity of the data can be achieved by ensuring that MANET often suffers from security attacks because of its features like open medium, changing its topology dynamically, lack of central monitoring and management, cooperative algorithms, and no clear defense mechanism. These characteristics have changed the challenging situation of the MANET against the security threats [3, 4]. Detecting these intrusions is a difficult procedure that is full of uncertainty and indeterminacy problems. Neutrosophic [10] discipline provides a logical base for dealing with uncertainty problems. In neutrosophic theory, truth-membership, falsity, and indeterminacy values are independent and calculated separately. Hence, the indeterminacy of the MANET attack could be identified through the intrusion detection process. This research is interested in integrating the neutrosophic concepts with the artificial intelligence search capabilities to produce an accurate neutrosophic intrusion detection system (NIDS).

In literature, many researchers were concerned with the security of the MANET. They proposed many IDS to detect security threats and find a model to prevent their consequences. For example, Ektefa [15] compared C4.5 and SVM to show the performance of both algorithm and FAR values too. Among these two C4.5 works better compared to other. Since the performances of a classifier are often evaluated by an error rate and it does not suit the complex real problems, in particular multiclass, Holden [16] has proposed hybrid PSO algorithm that can deal with nominal attributes without going for the both conversion and nominal attribute values. To overcome the drawback (features) that the PSO/ACO algorithm lacks, the proposed method shows simple rule set efficiently to increase in accuracy. Likewise

```
Begin
      Initialize a population
      Evaluate fitness for each individual
      While (! stop condition) do
            Proceed the cross over process.
            Proceed the mutation process.
            Evaluate new individual.
            Select individual to replace and their replacement.
            Update stop condition.
      End
      Return best solution
      End
```

ALGORITHM 1

Ardjani [17] applied SVM with PSO as (PSO-SVM) to optimize the performance of SVM. 10-fold cross-validation is done to estimate the accuracy. It utilizes the advantage of minimum structural risk with global optimizing features. The result shows better accuracy with high execution time. Since [19] is an existence of multidimensional data set, it is necessary to extract the features and also to remove the redundant and inconsistent features that affects classification. Based on this, information gain and genetic algorithm have been combined to select the significant features. This method shows better accuracy when features are selected than individually applied.

2.2. Genetic Algorithms. Genetic algorithms (GAs) [14, 20–22] utilize the Darwin's evolution theory of life. Massive machines [23] and transporters [1] are obvious and frequent applications of the GAs. Through the progress of the GA, the fit individuals are passed to the next generation. This guarantees the survival and reproduction of the best individuals over the worst. Upon assigning a problem, GAs evolve to find the optimal solution. The process begins with a population of random collection of solutions (chromosomes). During selection, the new population is formed from the best individuals. The individual fitness should be the criteria of the selection. The chance of reproducing fit individuals is guaranteed. The fitness function is defined by the issued problem. The new population (having the best individuals) goes through the crossover or mating process. The motivation of producing better populations during the GA process is the obvious hope of finding the optimal solution. The selection-reproduction processes are repeated until the optimal solution is found or the end of iterations is reached with an acceptable error rate. The algorithm is declared in Algorithm 1.

3. Designing the Proposed Method

Intrusion Detection System (IDS) is an essential security part for any online network nowadays. An intrusion is "a collection of actions that try to comprehend the privacy, integrity or availability for various resources." Intrusion can likewise be characterized as "a collection of actions imagine to get unapproved assets, abuse rights, cause finish frameworks and systems smashed, diminish running intensity, or refuse any

assistance." In this manner, IDS might be a framework to monitor events in PCs or systems and examinations and check the frameworks uprightness and privacy. IDS could be figured as an arrangement of if-then rules that depict the potential intrusions of the network or systems. Finding the ideal arrangement of these rules is a vital search problem. These algorithms switch the problem in a particular space into a model by utilizing a chromosome-like data structure and develop the chromosomes using selection, recombination, and mutation operators. GA is used as critical solving strategy and gives ideal solution of the problem GA works on the Darwinian principle of reproduction. It is a changed set of individual objects, in which each correlating fitness value goes into new generation of population and after that applies crossover and mutation function. The proposed hybrid combines SOFM and GA algorithms to produce the neutrosophic rules in two phases. The first phase sets the neutrosophic variables by creating the membership, nonmembership, and indeterminacy functions for the neutrosophic subsets of the variables. The implementation for the first stage is done by using SOFM from a prior research [13]. The outcome of this stage is passed to GAs [4] along with the training data to first randomly generate initial population; thereafter the neutrosophic correlation coefficient is used as a fitness function to pick out the most fit rules with regard to the training data. Afterwards, the test data is utilized to check for the precision of the rules created.

3.1. Formatting Neutrosophic KDD Features Using SOFM. In order to build a neutrosophic IDS, the system should be based on neutrosophic variables. The regular features in the KDD data set cannot be used in neutrosophic processing. Hence, reformatting the KDD features into neutrosophic ones is a preprocessing step in the intrusion discovery system. Self-Organized Feature Maps (SOFM) are unsupervised artificial neural networks that were used to define the neutrosophic variables [13]. SOFMs capabilities cluster inputs using self-adoption techniques. These capabilities were utilized in generating neutrosophic functions for the subsets of the variables. The SOFMs are used to define the membership, nonmembership, and indeterminacy functions for the KDD data set features. The algorithm for generating the neutrosophic features definitions is cleared in Algorithm 2.

Input: input_data vectors(Training_data set), Input_dim, output_dim,
Output: neutrosophic variable membership, nonmembership and indeterminacy functions
//membership function generation
　　(1) Training_Data←Read_data(membership_data)
　　(2) Membership_data← SOFM(Trainig_data, Input_dim, output_dim)
　　(3) Draw (Membership_data)
　　(4) Training_Data←Read_data(nonmembership_data)
　　(5) Non_Membership_data← SOFM(Trainig_data, Input_dim, output_dim)
　　(6) Draw (Non_Membership_data)
　　(7) Indeterminancy← Calculate_ind(Membership_data, Non_Membership_data)
　　(8) Draw (Indeterminancy)
End

Function SOFM　　　　　　　　　　　　　　　**Function Update_weights**
Input: Trainig_data, Input_dim, output_dim　　　**Input**: Winning_neuron$\mathbf{q_j}$
Output: Output _Function　　　　　　　　　　　**Output**: Update_weights
　　Initialize_SOFM (input_neurons, output_neurons)　　　(1) Find (Winning_neuron$\mathbf{q_j}$)

　　Randomly_Initialize_SOFM_Weights ()　　　(2) $\eta_{qj}[t] = \begin{cases} \mu[t] & j\epsilon N_q \\ 0 & j \notin N_q \end{cases}$

　　While Error>threshold **Do**
　　Foreach Record in **Training_Data**　　　(3) $w_j[t+1] = w_j[t] + \eta_{qj}[t](x_n[t] - w_j[t])$
　　Input_Record ();
　　Winning_neuron$\mathbf{q_j}$=$\mathbf{q}(x_n) = \mathbf{min}_{\forall j}\|x_n - w_j\|$;　　　(4) Output (Update_weights)
　　Update_weights (Winning_neuron$\mathbf{q_j}$);　　　(5) End fun
　　Endforeach
　　Error =Calculate_ErrorRate ();
　　End while
　　Retrieving_phase ();
　　Output Function←Network_Weights)
　　End fun

ALGORITHM 2: SOFM algorithm for generating the membership, nonmembership, and indeterminacy functions of the neutrosophic variable.
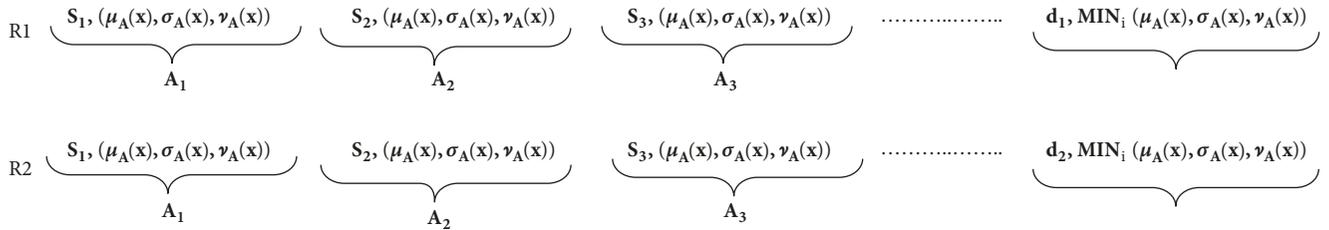

FIGURE 1: Neutrosophic rules of the knowledge based system.

These definitions are used during the GA search in fitness calculation (the neutrosophic correlation coefficient). The procedure of generating the neutrosophic IDS classification rules is introduced in the next section.

3.2. Creating Neutrosophic Rules Utilizing Genetic Algorithms. The neutrosophic knowledge based system is composed of a set of neutrosophic rules (Figure 1). This procedure is in charge of designing the neutrosophic conditional rules via applying the processing power of GAs at random initial population. After that, utilizing the neutrosophic correlation coefficient as a fitness function picks the most proper performers from the population. At the last stage, the population becomes the set of neutrosophic rules required for the neutrosophic inference engine of the IDS.

An individual represents the possible solution or the possible neutrosophic rule which is composed of a set of conditional propositions (neutrosophic attributes) and the consequence decision attribute. Each neutrosophic attribute and the decision attribute will occupy one gene within the individual. To demonstrate the neutrosophic format, each gene will be represented by $A \in S, (\mu_A(x), \sigma_A(x), \nu_A(x))$, where A is the neutrosophic feature that belongs to the subset S with degrees of membership $\mu_A(x)$, nonmembership $\nu_A(x)$, and indeterminacy $\sigma_A(x)$. The GA individual is presented in Figure 2.

Note that A1, A2, A3, and A5 are the neutrosophic attributes and s2, s3, s2, and s1 are the neutrosophic subsets of the attributes. $^\wedge$ is the logical and operator. Moreover, the previous neutrosophic conditional rule does not rely on A4

$A_1 \in S_2, (\mu_A(x), \sigma_A(x), \nu_A(x))$	$A_2 \in S_3, (\mu_A(x). \sigma_A(x). \nu_A(x))$	$A_3 \in S_2, (\mu_A(x), \sigma_A(x), \nu_A(x))$		$A_5 \in S_1, (\mu_A(x), \sigma_A(x), \nu_A(x))$	$d_1 \in S_3$

$$\text{If } (A_1 \in s_2 \wedge A_2 \in s_3 \wedge A_3 \in s_2 \wedge A_5 \in s_1) \text{ then } d_1 \in s_3$$

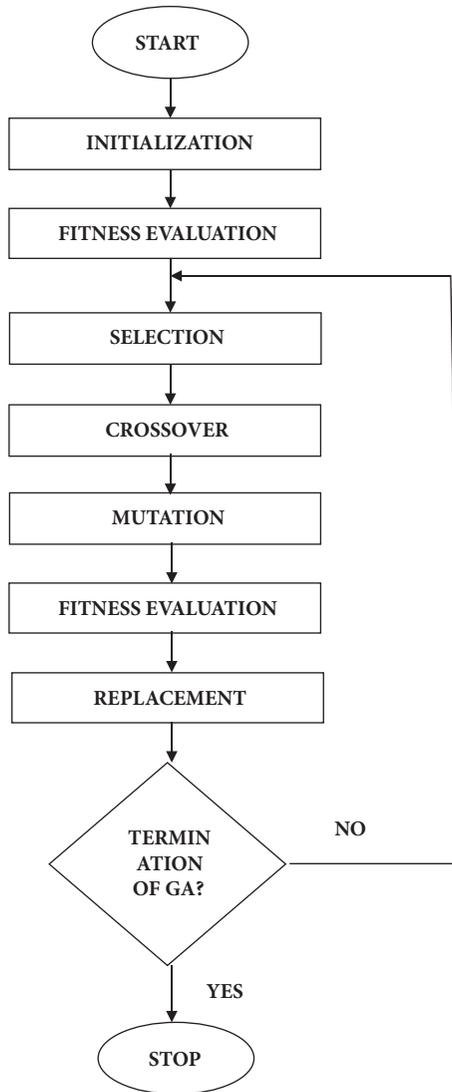FIGURE 2: The GA individual neutrosophic rules of a system has 5 feature and a dependent class.

FIGURE 3: Genetic algorithm process.

as a proposition and produces d1 as a decision within the neutrosophic subset s3. The flowchart for the generating neutrosophic rules procedure is provided in Figure 3. This procedure is composed of 7 phases with an iterated loop until the predefined number of iterations is performed. The first phase generates a collection of random neutrosophic rules according to the previous format (Figure 2) which forms the initial population. The second and third phases are the fitness evaluation and the selection process that determines the most fit individuals in the pool according to the neutrosophic correlation coefficient as a fitness function. The fourth phase is the crossover which produces new offspring from the fit individuals selected during the selection phase. The fifth phase is the mutation which switches one of the genes randomly to help increase the performance but this is accomplished under very rare circumstances. The sixth and seventh phases recalculate the fitness of the new offspring's and replace the children in place of their parents. These phases are repeated for a number of iterations from the third to the seventh. After completion, the final fit generation will form the set of neutrosophic conditional rules for the neutrosophic IDS. The GA procedure is straight forward and mostly used

in the same way for most of classification applications. The main module that differentiates one application from the other is the fitness function calculation. In order to integrate the neutrosophic concepts within the GA, the neutrosophic correlation coefficient is used to find the rules with the highest dependency between the neutrosophic conditional features and the decision attribute. The coefficient equations are illustrated in the next section.

3.3. Fitness Function. The neutrosophic correlation coefficient is used to measure the fit rules. The neutrosophic correlation coefficient measures the degree of relation between propositions and the decision attribute. The generated neutrosophic rules that maximize this correlation will be the most fit rules in the population and will be selected and passed to the next generation during the GA process. Some operations on neutrosophic sets were introduced and studied by Salama et al. in 2012 [12]. S and Y are two neutrosophic sets in a finite space x={x1, x2,..........., xn}; the correlation of neutrosophic sets S and Y is defined as follows:

$$C(S,Y) = \sum_{i=1}^{n} \lfloor (\mu_S(x_i) . \mu_Y(x_i) + \sigma_S(x_i) . \sigma_Y(x_i) + \nu_S(x_i) . \nu_Y(x_i)) \rfloor \quad (1)$$

and the correlation coefficient of S and Y is given by

$$R(S,Y) = \frac{C(S,Y)}{(T(S) . T(Y))^{1/2}} \quad (2)$$

where $T(S) = \sum_{i=1}^{n} \lfloor (\mu^2_S(x_i) + \sigma^2_S(x_i) + \nu^2_S(x_i)) \rfloor$ and $T(Y) = \sum_{i=1}^{n} \lfloor (\mu^2_Y(x_i) + \sigma^2_Y(x_i) + \nu^2_Y(x_i)) \rfloor$; $|R(S,Y)| \leq 1$.

'S' refers to the conditional propositions in a neutrosophic if-then rule like srv_serror_rate type, the Srv_serror_rate type, and the flag [13], where 'Y' refers to the class attributes like there is an attack or not.

4. Experimental Results

The proposed hybrid aims to identify the attacks that take place in the network to classify them correctly and increase the detection rate of attacks; Figure 4 indicates the whole system. The hybrid consists of a preprocessing step and two major phases. The preprocessing step utilizes the WEKA [24] data mining tool to get the most important attributes from the KDD-99 data set. The first phase is the neutrosophic variables definition which converts the normal data into neutrosophic variables utilizing the Self-Organized Features Maps (SOFM) [13]. The neutrosophic definition of the variables along with the training KDD-99 data is fed into the classification process. The second phase is the neutrosophic IDS building. The proposed system utilizes the evolutionary capabilities of the genetic algorithms (GA) to find the appropriate classification (if-then) rules. Simulation of the proposed system is implemented by C# environment on Dell Inspiron 15.6" Laptop-Intel Core i5, Memory (RAM): 8.00 GB, system type: 64-bit operating system, and Windows edition: windows 10.

TABLE 1: GA parameters in experiments.

cross over rate	0.6
mutation rate	0.90
number of population	500
number of iterations	50

The original data set is composed of 42 attributes; hence a reduction preprocessing step is required. The preprocess stage is implemented using Waikato Environment for Knowledge Analysis (Weka) [24]. The reduction algorithm used is the Attribute Evaluator 'FuzzyRoughSubsetEval' and search method 'HillClimberWithClassifier.' After the reduction process, KDD-99 data file contains 25 features and 1721 instances as in Figure 5 in which red color is abnormal abnormal (attacks) and the blue is normal.

Through the neutrosophic variable definition phase, the SOFM is applied to the KDD-99 data set to define the MEMEBERSHIP, INDETERMINACY, and NONMEMEBERSHIP functions. The technique used for neutrosophic variables definition is illustrated in our previous work in [13]. Figure 6 shows the result of the neutrosophic features definition phase.

During the second phase, the IDS classification pattern which detects threats in the MANET network is implemented by an artificial intelligent algorithm (GA). Each feature from the KDD-99 data set will have three different definitions for MEMEBERSHIP, INDETERMINACY, and NONMEMEBERSHIP assumption for the variable values. These features along with a random subset of KDD-99 (training data) are passed to GA program to build the set of if-then rules which represent the neutrosophic IDS. The generated output file contains the most fit rules according to the steps in GA pseudo code. The fitness function of the GA is the neutrosophic correlation coefficient which is calculated according to (2). The GA program simulation is implemented in 3 dimensions "MEMEBERSHIP, INDETERMINACY, and NONMEMEBERSHIP" compared with other techniques or algorithms using only two dimension "MEMEBERSHIP and NONMEMEBERSHIP." The genetic algorithm parameters like crossover rate, mutation rate, number of population, and number of iterations are assigned to the values illustrated in Table 1.

At the end of the GA iterations, the final file will contain the most fit (if-then) rules. Each one will have MEMBERSHIP, INDETERMENANCY, and NONMEMEBERSHIP values for each feature. These rules will be the inference engine for the neutrosophic IDS. The inference methodology used mimics the (min-max) Mamdani inference methodology [25]. The output rules files generated have the most appropriate neutrosophic rules which indicate whether a given instance is a normal connection or an attack. On applying a new instances (network data) to the system, the program selects the MIN $(\mu_A(x), \sigma_A(x), \nu_A(x))$ value from all features in each rule in assumption that all features are interconnected by 'AND' gate. Also, assuming that all rules are connected by 'OR' gate, the program selects the rule with MAX $(\mu_A(x), \sigma_A(x), \nu_A(x))$ value to be the matched one.
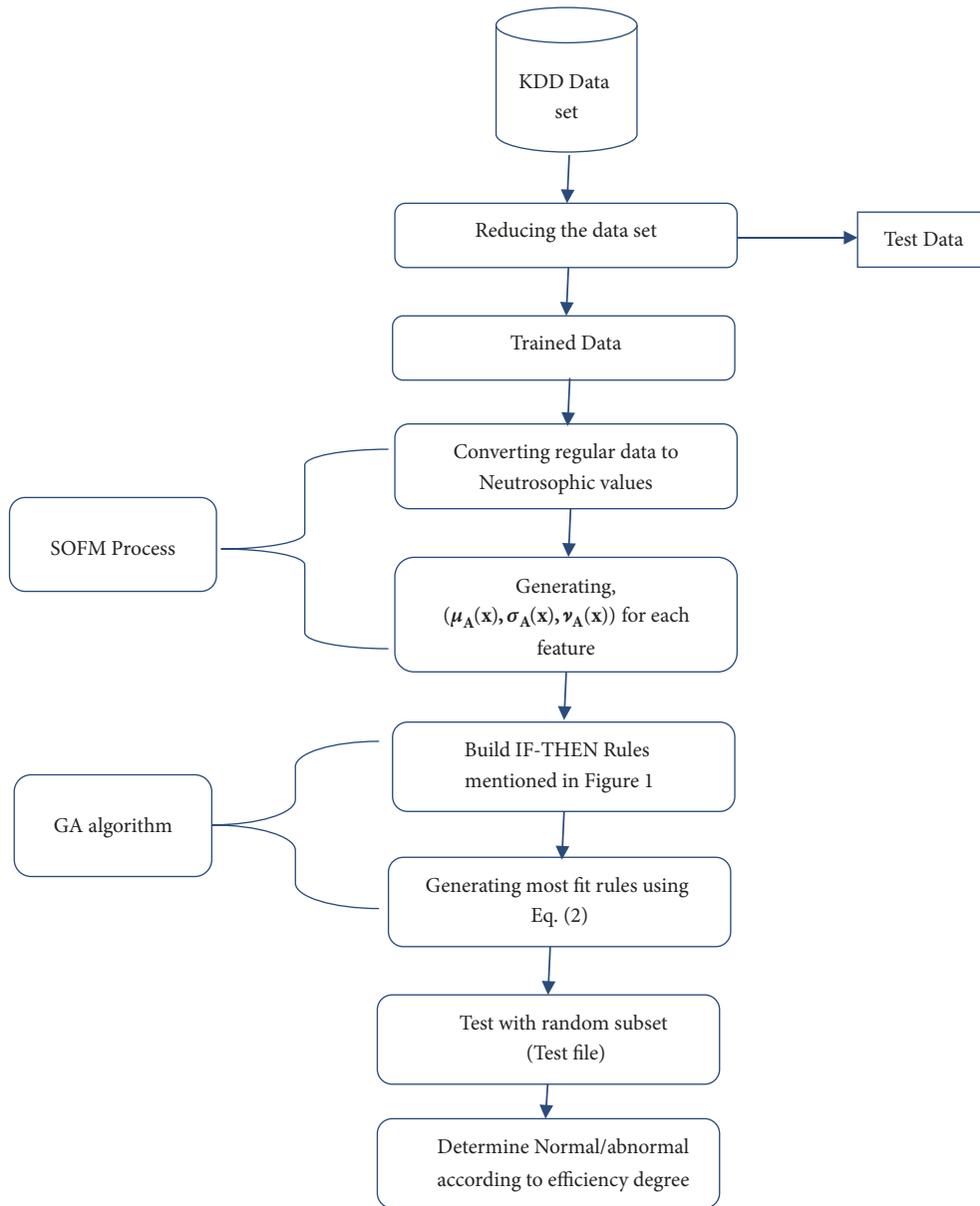
Figure 4: A frame work design for the whole system.

Then, the result of that matched rule will be compared to the actual KDD-99 data set to calculate the number of accurate instances percentage and the false rate percentage. During experiments, the KDD99 data set is divided randomly into two equal data sets (training and testing). The neutrosophic IDS is built using the training data set, and then its accuracy is measured by the new instances in the test data set. The neutrosophic IDS reached an average accuracy 99.3608% which indicates that the proposed technique is more accurate than the previous algorithms used in this area [20, 26, 27]; this appears in Table 2 and Figures 7 and 8. The results shown in Table 1 represent the accuracy, sensitivity, and specificity values for the proposed neutrosophic genetic algorithm against C4.5, SVM, C4.5 +ACO, SVM + ACO, EDADT, SVM + PSO, and C4.5 + PSO algorithms [18]. In light of values obtained, the precision of C4.5 is 93.23%, the precision of SVM is 87.18%, the precision of C4.5 + ACO is 95.06%, the precision of SVM + ACO is 90.82%, the precision of C4.5 + PSO is 95.37%, the precision of SVM + PSO is 91.57%, and the accuracy of Improved EDADT is 98.12%. It is obvious that the neutrosophic IDS generated by GA takes highest precision percentage when compared to all seven classification based algorithms. Figure 7 indicates the corresponding chart for the result obtained in Table 2. Figure 8 shows the performance of existing and proposed neutrosophic intrusion detection system (IDs) algorithm based on false alarm rate (FAR). Thus the proposed neutrosophic intrusion detection system (IDS) Algorithm effectively detects attack with less false alarm rate.
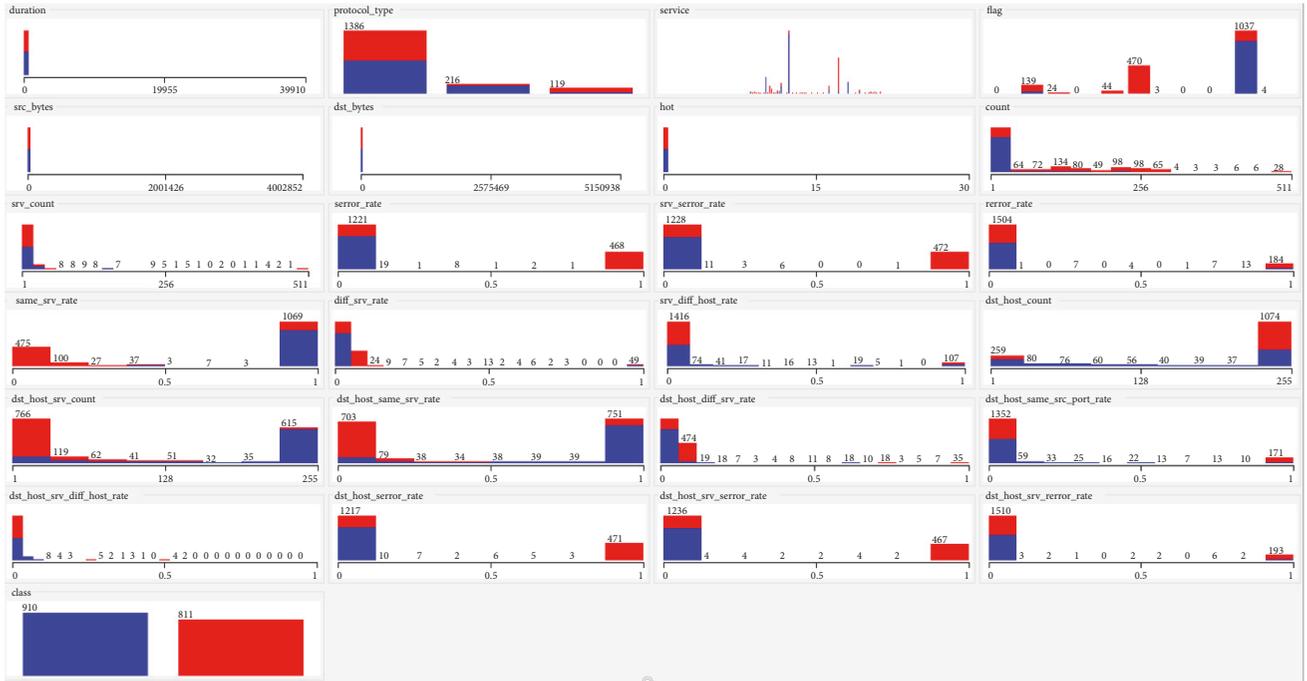
FIGURE 5: 25 features red color abnormal (attacks) and the blue is normal.

TABLE 2: Performance of proposed neutrosophic genetic algorithm vs. existing algorithms.

Algorithms	Accuracy (%)	FAR (%)
C4.5	93.23	1.65
SVM	87.18	3.2
C4.5+ACO	95.06	0.87
SVM+ACO	90.82	2.42
C4.5+PSO	95.37	0.72
SVM+PSO	91.57	1.94
EDADT	98.12	0.18
proposed neutrosophic genetic algorithm	99.3608	0.089

5. Conclusion and Future Work

The nature of the Mobile Adhoc Networks (MANETs) puts them under attacks from inside and outside the network. The security has two mechanisms: first preventing mechanisms like cryptography and hash function and second reactive mechanism like intrusion detection system. Such systems are full of indeterminacy and uncertainty. Hence, building a classification pattern for the neutrosophic variables to detect threats in MANETs is a vital purpose. This research indicates the integration of neutrosophic correlation coefficient into the genetic algorithm for upgrading an effective intrusion detection system. The proposed hybrid can increase the detection rate and reduce the false alarm rate in MANETs networks. Our experimental results prove that the proposed algorithm solves and detects the attacks in an effective manner compared with other existing works. Therefore, it will pave the way for an effective means for intrusion detection with better accuracy and reduced false alarm rate. In future, the performance of the system can be utilized in increasing the security of the system and predicting the intruders in MANETs networks by enhancing issues such as lack of resource consumption information to achieve an automatic intrusion detection system.

Data Availability

The [Excel] data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

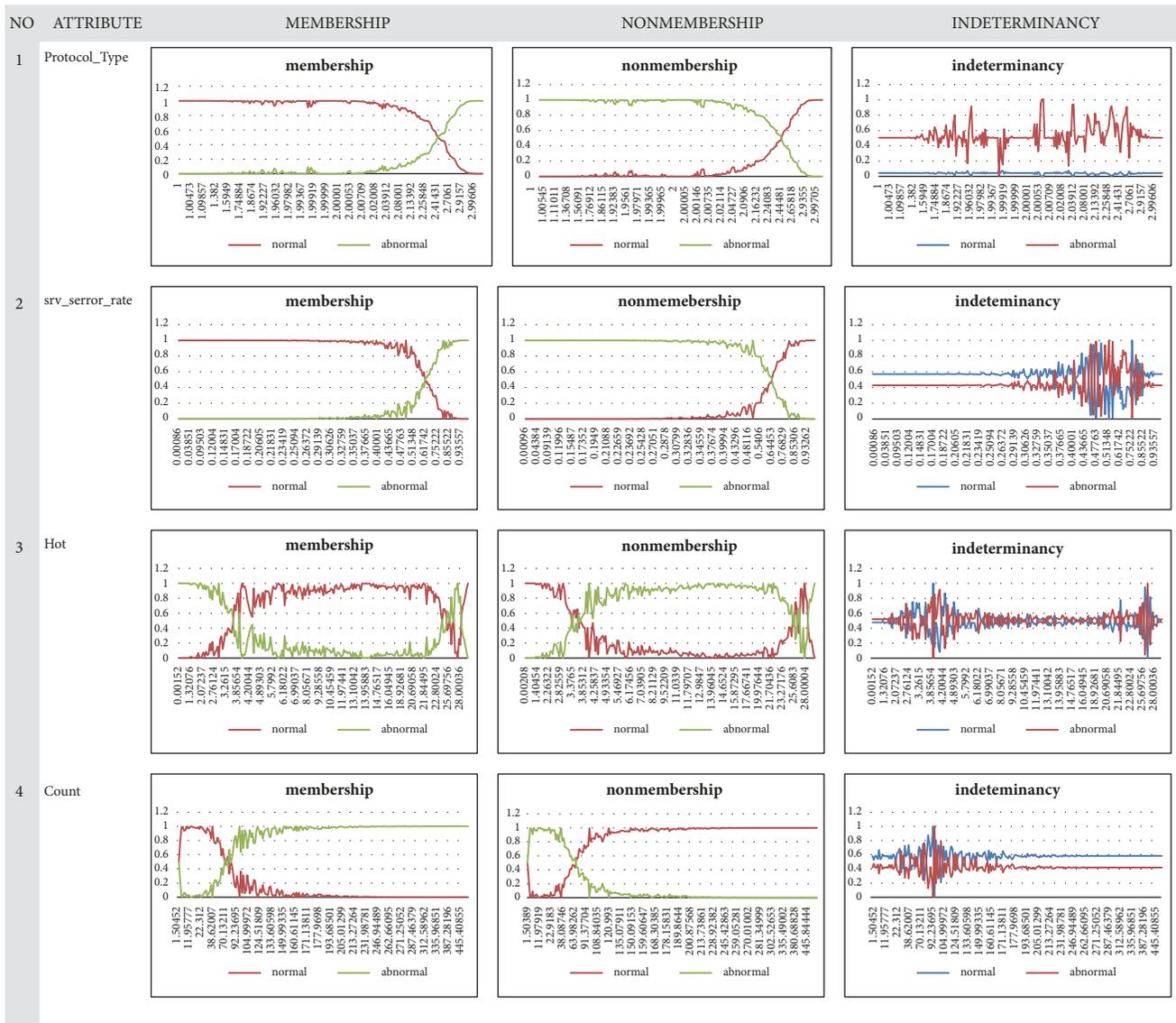The authors declare that there are no conflicts of interest regarding the publication of this paper.

FIGURE 6: the neutrosophic functions of membership, nonmembership, and indeterminacy for a number of KDD-99 data set.
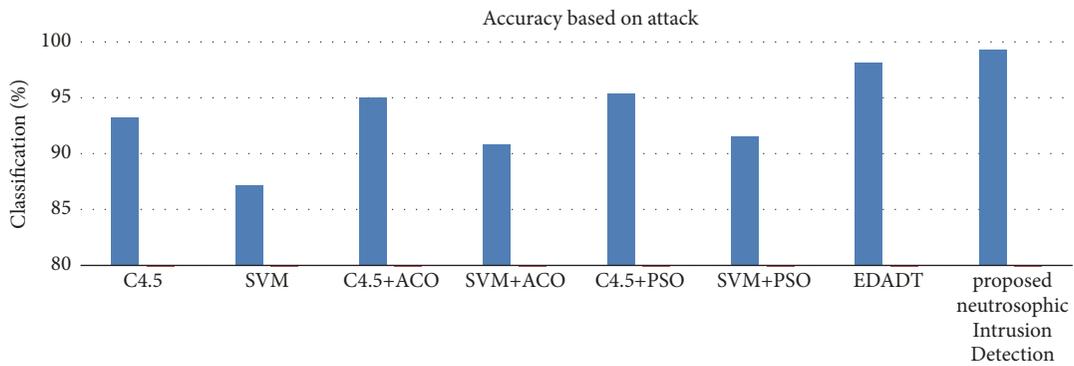

FIGURE 7: Results of neutrosophic genetic algorithm vs. existing algorithms.
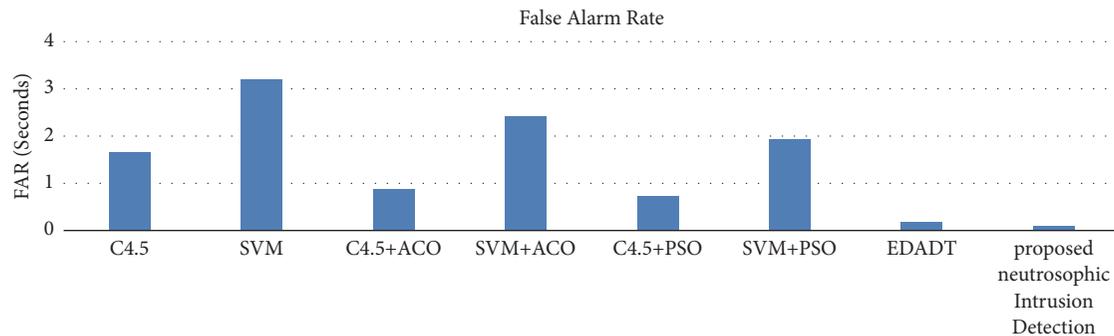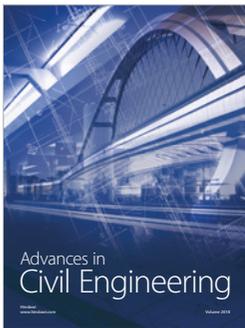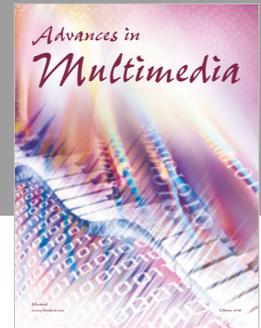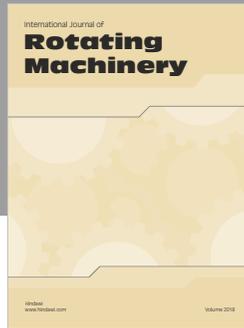
False Alarm Rate

FIGURE 8: False alarm rate of proposed neutrosophic genetic algorithm vs. existing algorithms.

Acknowledgments

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., NY, USA, 1995.

[2] D. P. Agrawal and Q.-A. Zeng, *Introduction to Wireless and Mobile Systems*, Brooks/Cole Publishing, 2002.

[3] H. Elwahsh, M. Hashem, and M. Amin, "Secure service discovery protocols for ad hoc networks," *Communications in Computer and Information Science*, vol. 131, pp. 147–157, 2011.

[4] K. Biswas and Md. L. Ali, *Security threats in Mobile Ad Hoc Network*, Blekinge Institute of Technology, 2007.

[5] P. V. Jani, *Security within Ad Hoc Networks, Position Paper*, PAMPAS Workshop, London, 2002.

[6] A. A. Salama, F. Smarandache, and S. A. ALblowi, "New Neutrosophic Crisp Topological Concepts, Neutrosophic Sets and Systems," 2014.

[7] A. A. Salama and F. Smarandache, *Neutrosophic Crisp Set Theory*, Educational Publishing, Columbus, Chesapeake, Ohio, USA, 2015.

[8] A. A. Salama, "Basic Structure of Some Classes of Neutrosophic Crisp Nearly Open Sets and Possible Application to GIS Topology," *Neutrosophic Sets and Systems*, vol. 7, pp. 18–22, 2015.

[9] S. Mitra and S. K. Pal, "Self-Organizing Neural Network As A Fuzzy Classifier," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 3, pp. 385–399, 1994.

[10] K. Atanassov, "Intuitionistic Fuzzy Set," *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87–96, 1986.

[11] "KDD Cup 1999 Data," https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[12] I. M. Hanafy, A. A. Salama, and K. Mahfouz, "Correlation of neutrosophic Data," *International Refereed Journal of Engineering and Science (IRJES)*, vol. 1, pp. 39–43, 2002.

[13] H. ELwahsh, M. Gamal, A. Salama, and I. El-Henawy, "Modeling Neutrosophic Data by Self-Organizing Feature Map: MANETs Data Case Study," *Procedia Computer Science*, vol. 121, pp. 152–159, 2017.

[14] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[15] M. Ektefa, S. Memar, F. Sidi, and L. S. Affendey, "Intrusion detection using data mining techniques," in *Proceedings of the International Conference on Information Retrieval and Knowledge Management*, IEEE, 2010.

[16] N. Holden and A. A. Freitas, "A hybrid PSO/ACO algorithm for discovering classification rules in data mining," *Journal of Artificial Evolution and Applications*, vol. 2008, Article ID 316145, 11 pages, 2008.

[17] F. Ardjani, K. Sadouni, and M. Benyettou, "Optimization of SVM multiclass by particle swarm (PSO-SVM)," in *Proceedings of the 2nd International Workshop on Database Technology and Applications, DBTA2010*, 2010.

[18] G. V. Nadiammai and M. Hemalatha, "Effective approach toward Intrusion Detection System using data mining techniques," *Egyptian Informatics Journal*, vol. 15, no. 1, pp. 37–50, 2014.

[19] N. Moustafa and J. Slay, "A Hybrid feature selection for network intrusion systems: Central points," in *Proceedings of the Australian Information Warfare and Security Conference*, pp. 5–13, Security Research Institute, Edith Cowan University, 2015.

[20] R. A. Becker, S. G. Eick, and A. R. Wilks, "Visualizing Network Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 1, pp. 16–28, 1995.

[21] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[22] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[23] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 227–261, 2000.

[24] "Machine Learning Group at the University of Waikato," https://www.cs.waikato.ac.nz/ml/weka/.

[25] M. Tavana, F. Azizi, F. Azizi, and M. Behzadian, "A fuzzy inference system with application to player selection and team formation in multi-player sports," *Sport Management Review*, vol. 16, pp. 97–110, 2013.

[26] E. Amiria, H. Keshavarzb, H. Heidaric, E. Mohamadid, and H. Moradzadehe, "Intrusion Detection Systems in MANET: A Review," *Procedia - Social and Behavioral Sciences*, pp. 453–459, 2014.

[27] R. Thanuja and A. Umamakeswari, "Effective intrusion detection system design using genetic algorithm for manets," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, 2016.

Journal of
Engineering

The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

Advances in
Multimedia

Advances in
Civil Engineering

Journal of
Control Science
and Engineering

Journal of
Robotics

Journal of
Electrical and Computer
Engineering

Advances in
OptoElectronics

VLSI Design

International Journal of
Navigation and
Observation

Modelling &
Simulation
in Engineering

International Journal of
Aerospace
Engineering

International Journal of
Chemical Engineering

International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components

Shock and Vibration

Advances in
Acoustics and Vibration

Hindawi

Submit your manuscripts at
www.hindawi.com