A Real Time Approach on Genetically Evolving Intrusion Detection using Neutrosophic Logic Inference System

S.Saravanakumar
Assistant Professor(IT), Sankara College of Science and Commerce

Abstract**:** *In this paper, we present an overview of our research in real time Neutrosophic logic based intrusion detection systems (IDSs). We focus on issues related to deploying a data mining-based IDS in a real time environment Information security has become a critical issue with the rapid development of business and other transaction systems over the internet. One of the toughest disputes in IDS is uncertainty handling. IDS offer a new challenge in handling uncertainty when normal and the abnormal behaviors in networked computers are hard to predict as the boundaries cannot be well defined. In this paper we have introduced a genetically evolving approach for IDS using Neutrosophic Logic classifier which is a generalization of the fuzzy logic, intuitionistic logic, and the three-valued logics that use an indeterminate value. The proposed method tripartition the incoming packets into normal, abnormal and indeterministic based on the degree of membership of truthness, degree of membership of indeterminacy and degree of membership of falsity. In this paper, we present an overview of our research in real time with Neutrosophic-based intrusion detection system. The rules generated by the Neutrosophic logic classifier is optimized using improvised genetic algorithm for better classification. This paper exhibit the efficiency of handling uncertainty in a real time environment precisely using Neutrosophic Logic Classifier based IDS. This proposed work significantly increases the detection rate and minimize the false alarm rate than the existing fuzzy based approaches.*

Keywords: *indeterministic, uncertainty, Neutrosophic, intrusion, genetic algorithm*

I. INTRODUCTION

As the world becomes more connected to the cyber world, attackers and hackers are becoming increasingly sophisticated, especially in the use of automated tools to penetrate systems. At the same time, cyber criminals are becoming more organized and can engineer highly coordinated and intricate attacks. Intrusion Detection Systems (IDS) have grown to be very popular in recent years. This is due to the obvious explosive growth of the Internet and the fact that most, if not all, of the most sensitive data is kept online [1].

Present available IDS generate significantly high number of false alarms. Therefore, the need of an alternative technique to minimize false alarms arises. Collecting these warning alarms and altering the intrusion detection system will increase resistance to attack.

One major challenge in ID is to identify the camouflaged intrusions from a huge amount of normal communication activities. It is highly demanding to apply data mining techniques to detect various intrusions. The biggest challenge of using data mining approaches in intrusion detection is that it requires a large amount of audit data in order to compute the profile rule sets. The Data Mining process requires high computational cost when dealing with large data sets. Reducing dimensionality can effectively cut this cost. Hence, dimensionality reduction is vital when data mining techniques are applied for intrusion detection.

Another toughest challenge in IDS is uncertainty handling. The normal and the abnormal behaviours in networked computers are hard to predict as the boundaries cannot be well defined. The prediction of the normal or abnormal behaviours is done by the comparison with predefined classes to find the most similar one. This prediction process may generate false alarms in many anomaly based intrusion detection systems. Consequently, it is observed that there is a fair chance of the existence of a non-null hesitation part at each moment of evaluation of an unknown object. At the same moment fine tuning the classification rules generated is the key to this challenging problem.

In this paper to overcome the problem of uncertainty in IDS we have adopted a new technique known as Neutrosophic Logic (NL) which is a generalization of the classical, three-valued and fuzzy logics. The goal of this approach is to classify patterns of the system behaviour in three categories (normal, abnormal and indeterministic). This NL can reduce the false signal rate in discovering intrusive behaviours. The rules generated by the NL are fine tuned using improvised genetic algorithm in order to obtain better results.

The subsequent sections of this paper are organized as follows; Section 2 describes the related work in the field of Intrusion Detection System, Section 3 deals with dataset description, Section 4 explains the basic concept of Neutrosophic Logic in detail, the section 5 explains the how Neutrosophic Logic classifiers used in intrusion detection. In section 6 the proposed approach to solve the problem of uncertainty is presented. Section 7 describes experiments and analysis of results and finally section 8 draws conclusion.

II. RELATED WORK

In the recent past there has been a growing recognition of deploying intelligent techniques for the creation of efficient and reliable intrusion detection systems. A significant challenge in providing an effective defence mechanism to a network perimeter is having the ability to detect intrusions and implement counter measures. Intrusion detection Techniques have been investigated since the mid 80s and depending on the type and source of the information used to identify security breaches, they are classified. A complete survey of these techniques is hard to be present at this point since there are more than 100 IDS based on machine learning techniques. Some of the best performed techniques are discussed in this section.

One of the most commonly used approaches in expert system based intrusion detection systems is rule-based analysis using Denning's [2] profile model. Rule-based analysis relies on sets of predefined rules that are provided by an administrator or created by the system. Unfortunately, expert systems require frequent updates to remain current. Earlier studies have utilized a rule-based approach for intrusion detection, but had a difficulty in detecting new attacks or attacks that had no previously described patterns [3]. Lately, the emphasis is being shifted to learning by examples and data mining paradigms. Neural networks have been extensively used to detect both misuse and anomalous patterns [4]. Recently, kernel-based methods, SVMs and their variants are being used to detect intrusions. Many researchers used data mining techniques to identify key patterns that help in detecting intrusions [5]. Distributed agent technology is being suggested by a few researchers to overcome the inherent limitations of the client–server paradigm and to detect intrusions in real time [6]

Typically, an IDS uses Boolean logic in determining whether or not an intrusion is detected and the use of fuzzy logic has been investigated as an alternative to Boolean logic in the design and implementation of these systems. Fuzzy logic addresses the formal principles of approximate reasoning [7]. It provides a sound foundation to handle imprecision and vagueness as well as mature inference mechanisms using varying degrees of truth. Since boundaries are not always clearly defined, fuzzy logic can be used to identify complex pattern or behaviour variations. This is accomplished by building an Intrusion Detection System that combines fuzzy logic rules with an expert system in charge of evaluating rule truthfulness.

Fuzzy Association rule are used to explore the possibility of integrating the fuzzy logic with Data Mining methods using Genetic Algorithms for intrusion detection [8]. A technique to generate fuzzy classifiers using genetic algorithms that can detect anomalies and some specific intrusion using two rules for normal class and other for the abnormal class are proposed using evolved fuzzy classifiers in intrusion detection .In [9], the authors established an anomaly detection model that integrated the association rules and frequency episodes with fuzzy logic to produce patterns for intrusion detection.

In [10], the authors developed an anomaly intrusion detection system combining neural networks and fuzzy logic. In [11], the authors applied genetic algorithms to optimize the membership function for mining fuzzy association rules.

In standard set theory each element is either completely in or not in a set this leads to the problems in case of vague concepts. Recent works deal with the fuzzy system but they fail to solve the indeterminacy (unknown) problem. In this proposed research work, to overcome the problem of uncertainty in IDS we have adopted a new technique known neutrosophic Logic which is a generalization of the classical, three-valued and fuzzy logics. The goal of this approach is to classify patterns of the system behaviour in three categories (normal, abnormal and indeterministic). This approach can reduce the false signal rate in discovering intrusive behaviours.

III. AN INTRODUCTION TO NEUTROSOPHIC LOGIC

3.1 Non-standard analysis

In 1960 Abraham Robinson has developed the non-standard analysis, a formalization of analysis and a branch of mathematics logic, which rigorously defines the infinitesimals. An infinitesimal is an infinitely small number. Let $\varepsilon>0$ be such an infinitesimal number. The hyper-real number set is an extension of the real number set, which includes classes of infinite numbers and classes of infinitesimal numbers. Let's consider the non-standard finite numbers 1+ = 1+ε, where "1" is its standard part and "ε" its non-standard part, and –0 = 0-ε, where "0" is its standard part and "ε" its non-standard part. Then, we call]–0, 1+[a non-standard unit interval. Obviously, 0 and 1, and analogously non-standard numbers infinitely small but less than 0 or infinitely small but greater than 1, belong to the non-standard unit interval.

3.2 Neutrosophic Logic

Smarandache[12] extended neutrosophy to neutrosophic logic, neutrosophic sets, and so forth. In bivalent logic, the truth value of a proposition is given by either one (true) or zero (false). NL is a multi-valued logic, in which the truth values are given by an amount of truth, an amount of falsehood and an amount of indeterminacy [13]. Each of these values is between 0 and 1. In addition, the values may vary over time, space, hidden parameters, etc. Further these values can be ranges.

NL which is a non standard analysis of tripartition such as degree of membership of truthness T, degree of membership of indeterminacy I and degree of membership of falsity F.

- ❖ To maintain the consistency with the classical and fuzzy logics and with probability there is the special case where T+I+F = 1.
- ❖ But to refer to Intuitionistic logic, which means incomplete information on a variable proposition or event one has T+I+F < 1.
- ❖ Analogically referring to Paraconsistent logic, which means contradictory sources of information about a same logical variable, proposition or event one has T+I+F>1.

Thus the advantage of using NL is that this logic distinguishes in philosophy between relative path truth that is a truth in one or a few worlds only noted by 1 and absolute truth denoted by 1^+. Likewise NL distinguishes between relative falsehood, noted by 0 and absolute falsehood noted by $^-0$ in non-standard analysis

Compared to the Fuzzy Set, the Neutrosophic Set can discriminate between 'absolute membership' (appurtenance) of an element to a set ($T=1^+$), and 'relative membership' (T=1), whereas the 'partial membership' is represented by $0 < T < 1$. Also, the sum of neutrosophic membership components (truth, indeterminacy, falsehood) are not required to be 1 as in fuzzy membership components, but may be any number between 0 and 3.

Constants: (T, I, F) truth-values, where T, I, F are standard or non-standard subsets of the non-standard interval$]^-0, 1^+[$, where $n_{inf} = \inf T + \inf I + \inf F \geq {}^-0$, and $n_{sup} = \sup T + \sup I + \sup F \leq 3^+$.

The NL is a formal frame trying to measure the truth, indeterminacy, and falsehood.

IV. NEUTROSOPHIC LOGIC CLASSIFIERS FOR INTRUSION DETECTION

Essentially all the information in the real world is imprecise, here imprecise means fuzzy, incomplete and even inconsistent. There are many theories existing to handle such imprecise information, such as fuzzy set theory, probability theory, intuitionistic fuzzy set theory, paraconsistent logic theory, etc. These theories can only handle one aspect of imprecise problem but not the whole in one framework. For example, fuzzy set theory can only handle fuzzy, vague information not the incomplete and inconsistent information. In this proposed work, we unify the above-mentioned theories under one framework. Under this framework, we can not only model and reason with fuzzy, incomplete information but also inconsistent information without danger of trivialization. This framework is called Neutrosophic Logic (NL). NL was created by Florentin Smarandache (1995) and is an extension/combination of the fuzzy logic, intuitionistic logic, paraconsistent logic, and the three-valued logics that use an indeterminate value.

4.1 Relationship between Neutrosophic and other sets

1. The classical set, $I = \phi$, $\inf T = \sup T = 0$ or 1, $\inf F = \sup F = 0$ or 1 and $\sup T + \sup F = 1$
2. The fuzzy set, $I = \phi$, $\inf T = \sup T \in [0, 1]$, $\inf F = \sup F \in [0,1]$ and $\sup T + \sup F = 1$.
3. The interval valued fuzzy set, $I = \phi$, $\inf T$; $\sup T$; $\inf F$; $\sup F \in [0, 1]$, $\sup T + \inf F = 1$ and $\inf T + \sup F = 1$.
4. The Intuitionistic fuzzy set, $I = \phi$, $\inf T = \sup T \in [0, 1]$, $\inf F = \sup F \in [0, 1]$ and $\sup T + \sup F \leq 1$.
5. The interval valued intuitionistic fuzzy set, $I = \phi$, $\inf T$, $\sup T$, $\inf F$, $\sup F \in [0, 1]$ and $\sup T + \sup F \leq 1$.
6. The paraconsistent set, $I = \phi$, $\inf T = \sup T \in [0, 1]$, $\inf F = \sup F \in [0, 1]$ and $\sup T + \sup F > 1$.
7. The interval valued paraconsistent set, $I = \phi$, $\inf T$, $\sup T$, $\inf F$, $\sup F \in [0, 1]$ and $\inf T + \inf F > 1$.

The relationship among Neutrosophic set and other sets is illustrated in Fig 1. Note that in Fig. 1 such as a → b means that b is a generalization of a.

V. SYSTEM ARCHITECTURE OF REALTIME INTRUSION DETECTION USING NEUTROSOPHIC LOGIC INFERENCE SYSTEM

In this paper we proposed a new approach for detecting intrusion using Neutrosophic Logic Inference Engine and improvised genetic algorithm in real time. The idea of designing a flexible system named Real Time Genetically evolving Neutrosophic Logic Inference System based IDS (RT-GNID) was conceived for applying it in a real-time environment. The system was constructed to detect the malicious attacks in the real time network which are possible to take place beyond the certainty environment.

RT-GNID consists of five types of components which are discussed below:

Packet Capture - It is the beginning process in NIDS. It can be implemented but setting the working mode of the network card as the promiscuous mode. The network card under common mode can only receive the packet whose destination address is the network card itself. Those packets alone are not sufficient to serve fro the data source of the NIDS. So, it is necessary to set the network cards working mode as the promiscuous mode.

Under this mode the network card can receive not only packet sent to it but also the packet routed to some other hosts. It reads the raw network packets off the wire and stores them on the disk.

Network Data pre-processor- The collected raw data is pre-processed Data transformation such as normalization may improve the accuracy and efficiency of mining algorithms. Such methods provide better results if the data to be analysed have been normalized, that is, scaled to specific ranges such as [0.0, 1.0] . The attribute values of this dataset are differs with each other. Before utilizing the data it should be normalized which scaled to fall within a small, specified range of value because Neutrosophic Logic values falls between -0 to 1^+. The normalization used here is z-score normalization. Discretization also performed to reduce the number of values of continuous attribute by dividing the range of the attribute into intervals. Interval labels were used to replace actual data values.

Z-score normalization

In z-score normalization (or zero-mean normalization), the values for an attribute A are normalized based on the mean and standard deviation of A. A value v of A is normalized to v' by computing

$$\text{v} = \frac{v - \overline{A}}{\sigma_A}$$

where $\overline{A}$ and σ_A are the mean and standard deviation, respectively, of attribute A. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers which dominate the min-max normalization.

Feature Extraction: Feature selection and extraction is one of the pivotal problems in implementing the IDS. Most of the existing IDs use all the features in the network packet to evaluate and look for known intrusive patterns. Some of these features are irrelevant and redundant. The drawback of this approach is a lengthy detection process and degrading performance of an ID system. Some of these features are irrelevant and redundant. In this Dimensionality reduction of the attributes is performed using Best First Search which extracts the essential attributes from each TCP header:

1. Source IP address (src)
2. Destination IP address (dest),
3. Source port number (sport) ,
4. destination port number (dport) ,
5. tcp control bits (tcpflag),
6. packet data length (len),
7. The date and time the packet was sent.

Neutrosophic logic Inference Engine - A Neutrosophic classifiers for solving the three class classification problem is used in this propose approach to generate a set of three rules, one for normal class, next one for the abnormal class and the last for indeterministic class, where the condition part is defined using only the monitored parameters and the conclusion part is an atomic expression for the classification attributes.

Fine tuning rules- The rules generated by neutrosophic logic inference engine has to be fine tuned in order to produce best result. The problem of ambiguity among the generated rules are overwhelmed by implementing Improvised Genetic Algorithm

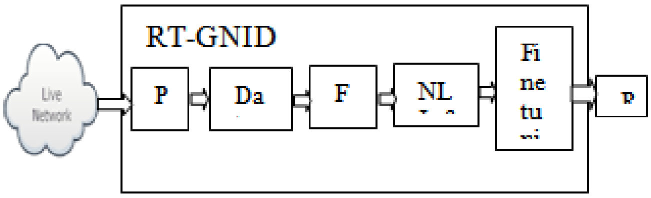
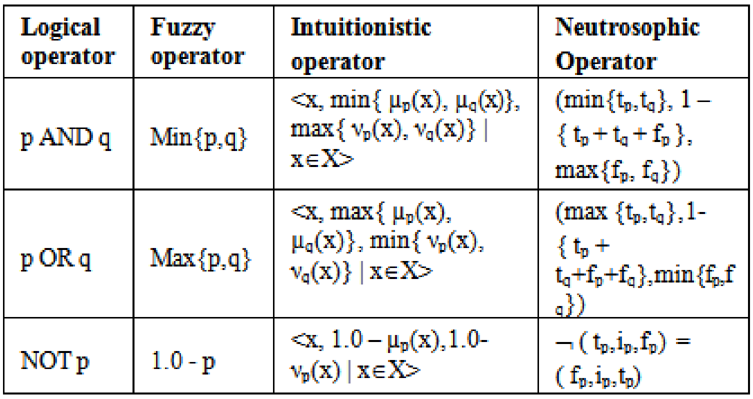


Fig 1 Overview of Proposed architecture of RT-GNID

The figure 1 depicts the overview of the proposed architecture in real time environment
The proposed method IDS is as follows:
Step 1: Collecting the dataset from packet capture
Step 2: Dataset pre-processing using Z-Score which normalize the dataset
Step 3: Feature selection using best first search method for finding potential attributes
Step 4: Adopting NLC for classifying the dataset into three classes namely normal, abnormal and indeterministic
Step 5: The rules generated by NLC are codified in the format of chromosome using complete tree representation
Step 6: Improvised Genetic Algorithm is applied on codified rules to yield best rules for classification.
Step 6: After tuning the rules, the testing datasets are validated.

VI. NEUTROSOPHIC LOGIC AND THREE CLASS CLASSIFICATION

Table 3 Neutrosophic Logic Operator



Logical operator	Fuzzy operator	Intuitionistic operator	Neutrosophic Operator
p AND q	Min{p,q}	$<x, \min\{ \mu_p(x), \mu_q(x)\}, \max\{ \nu_p(x), \nu_q(x)\} \mid x \in X>$	$(\min\{t_p, t_q\}, 1 - \{ t_p + t_q + f_p \}, \max\{f_p, f_q\})$
p OR q	Max{p,q}	$<x, \max\{ \mu_p(x), \mu_q(x)\}, \min\{ \nu_p(x), \nu_q(x)\} \mid x \in X>$	$(\max \{t_p, t_q\}, 1- \{ t_p + t_q + f_p + f_q\}, \min\{f_p, f_q\})$
NOT p	1.0 - p	$<x, 1.0 - \mu_p(x), 1.0- \nu_p(x) \mid x \in X>$	$\neg (t_p, i_p, f_p) = (f_p, i_p, t_p)$

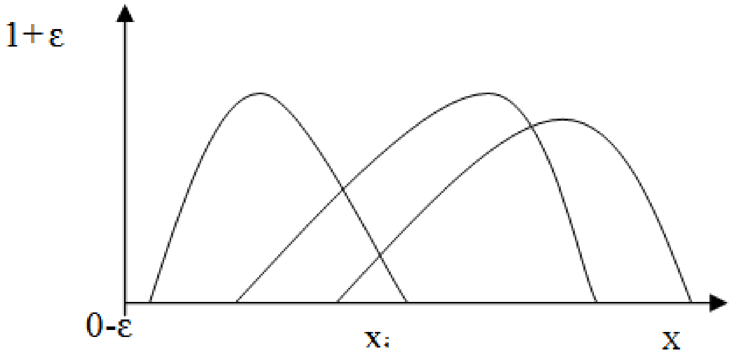


Fig2. Membership function of Neutrosophic Logic

In figure 2 the object x has denoted by the degree of membership of truth value, false value and indeterminacy. The NL allows an object to belong to different classes at the same time. This concept is helpful when the difference between the classes is not well defined. It is the case in the intrusion detection task, where the difference between the normal and abnormal classes is not well defined. Using these linguistic concepts atomic and complex NL expression can be built. An atomic neutrosophic expression is an expression

Parameter is [not] neutrosophic set
Where, Parameter is an object and neutrosophic set is a defined neutrosophic space for the parameter. The Truth Value (TV) of an atomic expression is the degree of membership of the parameter of the neutrosophic set. Because TV's are expressed by numbers between $^-0$ and 1^+.

Here $^-0$ means absolutely false, 1^+ means absolutely true, 0 means relative false and 1 means relative true and other value means partially true or false.

The neutrosophic expression evaluation process is reduced to arithmetic operations. Also, for each classical logic operator, fuzzy logic arithmetic operator, Intuitionistic logic operator, there is a common neutrosophic arithmetic operator which is shown in the table 3

Neutrosophic rules have the form

R: If Condition then Consequent [Weight]

Where,

- Condition is a complex neutrosophic expression(ie) that use NL and atomic neutrosophic expressions
- Consequent is an atomic expression
- Weight is a real number that defines the confidence of the rule

6.1 Neutrosophic Logic Classifiers and three classes classification problem

In this there are three classes where every object should be classified. These classes are called normal, abnormal and indeterministic. The dataset used by the learning algorithms consists of a set of object, each object with n + 1 attributes. The first n attributes define the object characteristics (monitored parameters) and the last attribute define the class that the object belong to (i.e) the classification attribute.

A Neutrosophic classifiers for solving the three class classification problem is a set of three rules, one for normal class, next one for the abnormal class and the last for indeterministic class, where the condition part is defined using only the monitored parameters and the conclusion part is an atomic expression for the classification attributes.

Some of the examples of neutrosophic rules for membership elements are as follows:

R_N: If x is high and y is low then
pattern is normal [0.3]

R_A: If x is medium and y is high then
pattern is abnormal [0.5]

$R_{I:}$ If x is medium and y is medium then
pattern is indefinite [0.2]

- R_N - is the rule for the normal class
- R_A – is the rule for abnormal class and
- R_I – is the rule for Indeterministic class

The Neutrosophic Logic rule truth-value is calculated as the product of the condition truth-value by the weight, i.e.

TV(R) = TV (Condition) * Weight

For membership function:

$TV(R_N)$ = TV (If x is high and y is low) * [0.3]

$TV(R_A)$ = TV (If x is medium and y is high) * [0.5]

$TV(R_I)$ = TV (If x is medium and y is medium) * [0.2]

Example:

For degree of membership of truthness, T(A):

$TV_{\mu(}t_{A)}(R_N)$ = TV (x is high and y is low) * (0.3)
= min{0.2, 0.7} * 0.3
= 0.2 * 0.3
= 0.06

$TV_{\mu(}t_{A)}(R_A)$ = TV (x is high and y is low) * (0.5)
= min{0.2, 0.7} * 0.5
= 0.2 * 0.5
= 0.10

$TV_{\mu(}t_{A)}(R_I)$ = TV (x is high and y is low) * (0.2)
= min{0.2, 0.7} * 0.2
= 0.2 * 0.2
= 0.04

For degree of membership of falsity, F (A):

$TV_{\mu(A)}(R_N)$ = TV (x is high and y is low) * (0.3)
= min{0.5, 0.2} * 0.3
= 0.2 * 0.3
= 0.06

$TV_{\mu(A)}(R_A)$ = TV (x is high and y is low) * (0.5)
= min{0.5, 0.2} * 0.5
= 0.2 * 0.5
= 0.10
$TV_{\mu(A)}(R_I)$ = TV (x is high and y is low) * (0.2)
= min{0.5, 0.2} * 0.2
= 0.2* 0.2
= 0.04

For degree of membership of Indeterminacy, I (A):

$TV_{\mu(A)}(R_N)$ = TV (x is medium and y is low) * (0.3)
= max {0.4, 0.2} * 0.3
= 0.2 * 0.3
= 0.06

$TV_{v(A)}(R_A)$ = TV (x is medium and y is medium) * (0.5)
= max {05, 0.4} * 0.5
= 0.5 * 0.5
= 0.25
$TV_{\mu(A)}(R_I)$ = TV (x is low and y is medium) * (0.2)
= max {0.1, 0.4} * 0.2
= 0.4 * 0.2
= 0.08

There are several techniques to determine the class that an object belongs to. One of these techniques is the maximum technique, which classifies the object as the class in the conclusion part of the rule that has the maximum truth-value, i.e.:

$$\text{Class} = \begin{cases} \text{N - If } TV(R_N) > TV(R_A) > TV(R_I) \\ \text{A - If } TV(R_A) > TV(R_N) > TV(R_I) \\ \text{I - If } TV(R_I) > TV(R_N) > TV(R_A) \end{cases}$$

Where,
N - Represents the Normal class,
A - Represents the Abnormal class and
I - Represents the Indeterministic class

VII. IMPROVISED GENETIC ALGORITHM

Grigorios N. Beligiannis et.al [14], proposed a new approach to improve the performance of classic genetic algorithm to achieve a better global exploration of the solution space while executing the minimum possible number of generations (function evaluations). This technique alleviates the enormous computational burden introduced by the local refining procedure, which is quite often useless in finding the optimal solution.

In their contribution, three different criteria for deciding when to apply restartings are proposed:

- Fitness function value
- Number of generations
- Mean fitness function value of population

7.1 Operator used in Genetic Algorithm Restartings

- ***Crossover operator****:* Suppose if s1 and s2 are two chromosomes then they are represented as
  - $S_1 = \{S_{11,} S_{12,} S_{13 \ldots\ldots} S_{1n}\}$,
  - $S_2 = \{S_{21,} S_{22,} S_{23 \ldots\ldots} S_{2n}\}$ are

Two chromosomes, select a random integer number $0 \le r \le n$, S_3 and S_4 are offspring of crossover(S_1, S_2),

- $S_3 = \{S_i \mid \text{if } i \le r, S_i \in S_1, \text{ else } S_i \in S_2\}$,
- $S_4 = \{S_i \mid \text{if } i \le r, S_i \in S_2, \text{ else } S_i \in S_1\}$

- ***Mutation Operator****: Suppose a chromosome $S_i = \{S_{11}, S_{12}, S_{13} \ldots\ldots S_{1n}\}$ Select a random integer number $0 \le r \le n$, S3 is a mutation of S_1,*
  - *$S_3 = \{S_i \mid if\ i \ne r,\ S_i \in S1_i, else\ S_i \in random\ (S_{1i})\}$*
- ***Selection Operator***: Suppose there are m individuals, we select [m/2] individuals and erase the others; the ones we select are having more fitness which means their profits are greater.
- ***Insertion Operator****:* Suppose there are m individuals, choose a constant number C having genomes of the new population and delete them. At the same time, choose a constant number C of random genomes of the old population and insert them into the new population.

VIII. IMPLEMENTATION OF NEUTROSOPHIC LOGIC CLASSIFIERS

In order to learn the Neutrosophic rules efficiently and design a compact and interpretable classification system we should concentrate in identifying best rule for accurate classification. Before making any prediction, every rule generated using Neutrosophic has to be evaluated to determine its prediction power. An expert's knowledge is used generally to construct a set of If –then Neutrosophic logic based statements to implement approximate reasoning. However in many cases the knowledge to elicit an optimized rule base is lacking.

To overcome this problem an Improvised Genetic Algorithm (IGA) is adopted in this proposed approach in order to remove redundant rules and detect the potential rules for optimized classification. The optimization problem is a three-goal objective function: maximize the sensitivity, maximize the specificity, and minimize the rule length.

Jonatan Gómez et al [15] proposed a new linear representation scheme for evolving fuzzy rules using the concept of complete binary tree structures. The same approach is adopted in this work for generating NL rules. Before applying IGA over the rules which is fetched from Neutrosophic logic space, the rule has to be converted to linear representation scheme with the help of complete expression tree.

To establish the process of linear representation we used the following grammar (in Backus Normal Form) for a free parenthesis logical expression:

<EXP> →<EXP><OPER><ATOMIC> | <ATOMIC>
<ATOMIC> →variable is [not] set
<OPER> →or | and

Applying repeatedly the previous definition, the following
Logical expression can be obtained:
If (src is .1 AND dest is .2 AND dport is HIGH AND tcpflag is .3 AND len is NOT HIGH AND time is LOW AND sport is NOT HIGH)
Then Pattern is normal [0.3]

In this way, the IGA for the normal class tries to develop a Neutrosophic logic rule. We evolve a rule for a specific class with one run of IGA. A Neutrosophic classifier can be represented by a set of m rules, where m is the number of different classes

R_1 : IF condition1 THEN data is $class_1$

…

R_m : IF condition1 THEN data is $class_m$

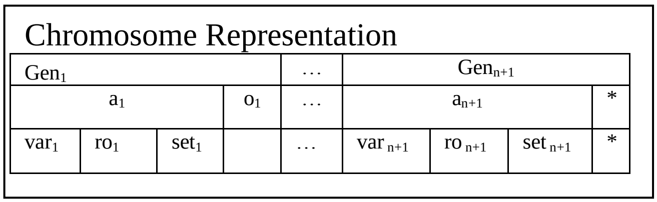
If m is the number of different classes, we run IGA m times. Only the condition part has to be codified as a linear chromosome with variable length, were leaf nodes are atomic expression and intermediate nodes is logical expression

With the help of complete expression tree the chromosome is defined as a set of n genes each is composed of an atomic condition

<Variable > is [not] < set >

and along with a logical operator

The logical expression is codified with n logic operators in a chromosome of n+ 1 gene, where ith gene is composed by the atomic expression a_i and the logic operator o_i. The last gene has an unused logic operator. The fig 3 shows the chromosome for a complete rule condition



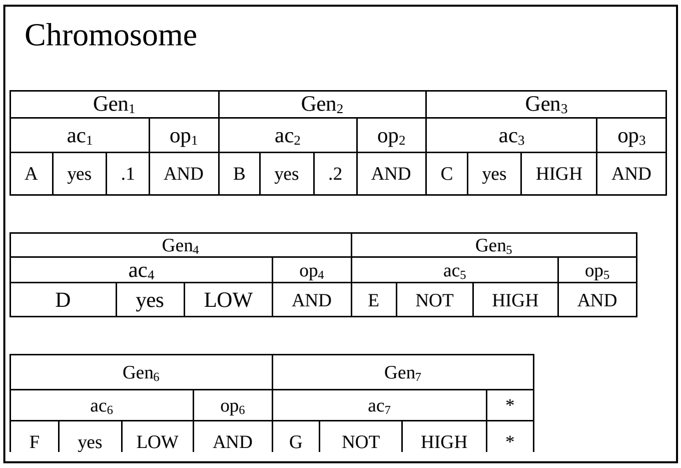
Chromosome Representation

Gen_1				…	Gen_{n+1}			
a_1			o_1	…	a_{n+1}			*
var_1	ro_1	set_1		…	var_{n+1}	ro_{n+1}	set_{n+1}	*

Figure 3 Chromosome representation

Fig 4 Representation of chromosome for a complete rule

For the expression the chromosome representation is shown in the figure 4



Chromosome

Gen_1				Gen_2				Gen_3			
ac_1			op_1	ac_2			op_2	ac_3			op_3
A	yes	.1	AND	B	yes	.2	AND	C	yes	HIGH	AND

Gen_4				Gen_5			
ac_4			op_4	ac_5			op_5
D	yes	LOW	AND	E	NOT	HIGH	AND

Gen_6				Gen_7			
ac_6			op_6	ac_7			*
F	yes	LOW	AND	G	NOT	HIGH	*

Fig 4 coding the expression A is .1 AND B is .2 AND C is HIGH AND D is .3 AND E is NOT HIGH AND F is LOW AND G is NOT HIGH

To implement IGA the condition part alone is codified as a chromosome as follows

A – Src, B- Dest, C – dport D- tcpflag E - len F - time G - sport

A is .1 AND B is .2 AND C is HIGH AND D is .3 AND E is NOT HIGH AND F is LOW AND G is NOT HIGH

IGA works in an iteration manner by generating new populations of strings from old ones. Every string is the encoded binary, real etc., version of a candidate solution. So Chromosome formatted above has to be represented in the form of binary string. If there are n variables then we use (log n/log 2) bits to represent each item

Consider the following example:
A is .1 AND **B is .2** AND **C is HIGH** AND **D is .3** AND **E is NOT HIGH** AND **F is LOW** AND **G is NOT HIGH**

Can be interpreted as
A1 AND **B1** AND **C1** AND **D1** AND NOT **E1** AND **F1** AND NOT **G1**

- ○ Here three bits strings represent variables, and one bit represent presence or absence of that variable
- ○ The relation operator part is codified with only *one* bit as the logic operator is either *and* or *or*. i.e1 – AND, 0 - OR
- ○ Here 000 – A1 , 001 – B1 , 010 – C1 , 011 –D1, 100 – E1 , 101 – F1 , 110 – G1

It is encoded as follows:

00011001110101101111110001101111100

These binary strings are used as the candidate solution for performing operation with genetic operators such as selection, crossover, mutation and insertion as discussed in the section 8 on an initially random population in order to compute a whole generation of new strings. IGA runs to generate solutions for successive generations. The probability of an individual reproducing is proportional to the goodness of the solution it represents. Hence the quality of the solutions in successive generations improves. The process is terminated when an acceptable or optimum solution is found.

8.1 Fitness Function Evaluation

We opt to seek the classification rule for each class separately because this leads to much faster and simpler search and has the potential to yield simpler rules this approach can leads to parallel processing of rules in the presence of many classes.
In this paper instead of using classical confusion matrix we introduced neutrosophic confusion matrix which corrects near misses in prediction by comparing the similarity of the predicted type of the actual type and giving credit for the similarity.
The fitness of a chromosome for the normal class is evaluated according to the following set of equations

$$\mathrm{TP} = \sum_{i=1}^{p} \mathrm{predicted}(\mathrm{normal\_data}_i)$$

$$\mathrm{TN} = \sum_{i=1}^{p} \begin{matrix}[1 - \mathrm{predicted}(\mathrm{abnormal\_data}_i) - \\ \mathrm{predicted}(\mathrm{indeterministic\_data}_i)]\end{matrix}$$

$$\mathrm{FP} = \sum_{i=1}^{p} \mathrm{predicted}(\mathrm{abnormal\_data}_i)$$

$$\mathrm{FN} = \sum_{i=1}^{p} \begin{matrix}[1 - \mathrm{predicted}(\mathrm{normal\_data}_i) - \\ \mathrm{predicted}(\mathrm{indeterministic\_data}_i)]\end{matrix}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Length} = 1 - \frac{Chromosomelength(rules)}{10}$$

Fitness = w1 * sensitivity + w2 * specificity + w3 * length

Here,

- p represents number of samples in the dataset used by each chromosome respectively.

- real is a function that returns when the data sample belongs to the training class and in other case.
- predicted is the IFS value of the condition part of the codified rule.
- TP means true positive, the outcome is correctly classified as positive.
- TN means true negative, the outcome is correctly classified as negative.
- FP means false positive, the outcome is incorrectly classified as positive
- FN means false negative, the outcome is incorrectly classified as negative when it is in fact positive.
- w_1, $w_{2,}$ w_3 are the assigned weights for each rule characteristics.
- Normal_data$_i$ is the subset of normal training patterns
- Abnormal_data$_i$ is the subset of abnormal training patterns and
- Indeterministic_data$_i$ is the subset of indeterministic training patterns.

By replacing abnormal/ indeterministic instead of normal in previous equation we can calculate the fitness for the abnormal and indeterministic class. The best chromosome in the population is chosen and the NL rule:

If <condition> then pattern is <class>

is added to the NLC. Here, <condition> is the condition represented by such gene, and <class> is the class pattern evolved by the improvised genetic algorithm.

IX. EXPERIMENTAL RESULT

Data set: The LAN was operated in a real environment which consists of 250 nodes and one server, blasted with multiple attacks. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted. Out of this database, a subset of 59261 data was used, of which 20% represent normal patterns. The data was partitioned into three different classes: normal, indeterministic and attack, where the attack is the collection of all different attacks belonging to the four classes. The objective of our proposed approach experiments is to separate normal and attack patterns. Data points were randomly generated which contain actual attacks and normal usage pattern

The effectiveness of handling incomplete and inconsistent information by NL leads to the construction of **RT-GNID** and tested their performance on the real time dataset. Dimensionality reduction, rules generation and fine tuning the generated rules are the three key steps in any intrusion detection system based learning algorithm.

In our work the Dimensionality of attributes are reduced using best first search which was adopted in our previous work. Our proposed model generates the detection rule based on the NLC. The main goal of this work is to generate fine NL rules to detect intrusions. Using improvised genetic algorithm the rules generated by the neutrosophic classifiers are fine tuned to produce best result. All the experiments were carried out on an Intel(R) Core(TM) i3processor, 2.13GHz PC with 4 GB RAM. The implementation is done using MATLAB Software.

A five fold validation was employed for [lim] evaluation. The dataset is divided into 2 parts. The dataset Training part includes 90% of all dataset and the testing part includes 10% of all dataset. The training dataset is used for acquiring rules and the testing dataset is used for validating rules. The process was repeated for five times and the score of the trained classifier was calculated as the average of twenty-five test applied.

Initially two hundred individual are chosen randomly from the chromosome population with the length of seven genes and maximum iteration are fixed to 200. After generating the initial population we used the fitness function as a metric to select the fit individuals. Three different fitness values are calculated for three classes (normal, abnormal and indeterministic) as mention in the section 8.1

An individual matches a class type when all the seven fields that constitute our search space of the individual match those of the class type. The rate of crossover was set to 0.6 (ie) five 200 individuals in any population 120 best individuals will be selected based on high fitness score and be made to undergo crossover to create offspring’s. We are exploring only seven fields, the crossover occurs only over these fields. Out of 120, the best 80 parents are then selected to complete the population size of 200. Thus the best fit parents also participate in the subsequent generations. The mutation rate has been fixed to 1% where in, only 2 individual out of a population size of 200 undergoes a change in one of the seven fields.

Thus the Evolution of NLC based Intrusion Detection System showed good performance in the increase of detection rate and reduces the false alarm significantly.

9.1 Results and Analysis

The false alarm rate and the accuracy rate are the two factors that define the cost function of an Intrusion Detection System. The average performance of RT-GNID proposed approach over twenty five test performed is shown in the table 6.

TABLE 6: COMPARISON OF THE PROPOSED MODEL

	Accuracy rate	False rate
DOS	98.3%	0.65%
R2L	96.1%	7.8%
U2R	89.2%	9.3%
Probe	94.7%	10.6%

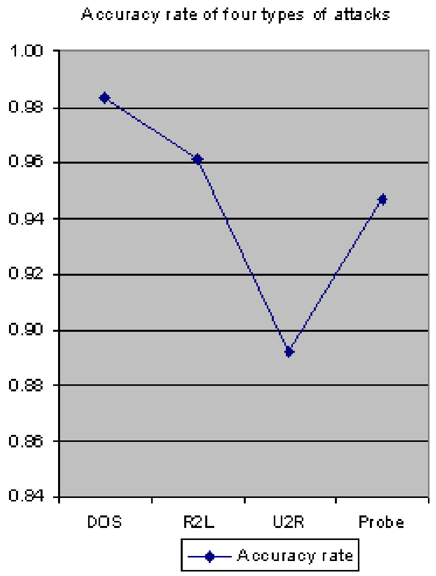


Fig 5 Accuracy rate of Proposed Model with four types of attacks

The aim of this research was to determine the maximum percentage of correctly classified instances. The NLC is a three class problem. We applied the roc curve analysis to evaluate the performance of the three different classifiers.

- Using simply the Neutrosophic rule for the normal class and varying a threshold (β) for the truth-value of the rule between $^{-}$0.0 and 1.0^{+}
- Using only the Neutrosophic rule for the abnormal class and varying a threshold (β) for the truth-value of the rule between -0.0 and 1.0+
- Using only the Neutrosophic rule for the indeterministic class and varying a threshold (β) for the truth-value of the rule between -0.0 and 1.0+.

The figure 5 and 6 depicts the accuracy rate and the false rate of the RT-GNID for dos, r2l, u2r and probe attacks separately.

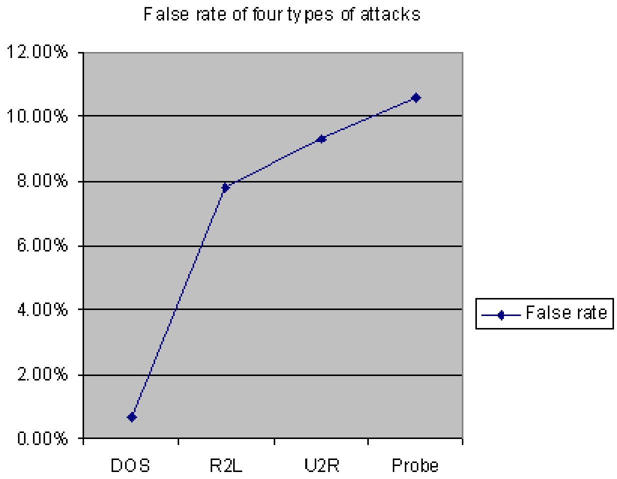


Fig 6 False rate of Proposed Model with four types of attacks

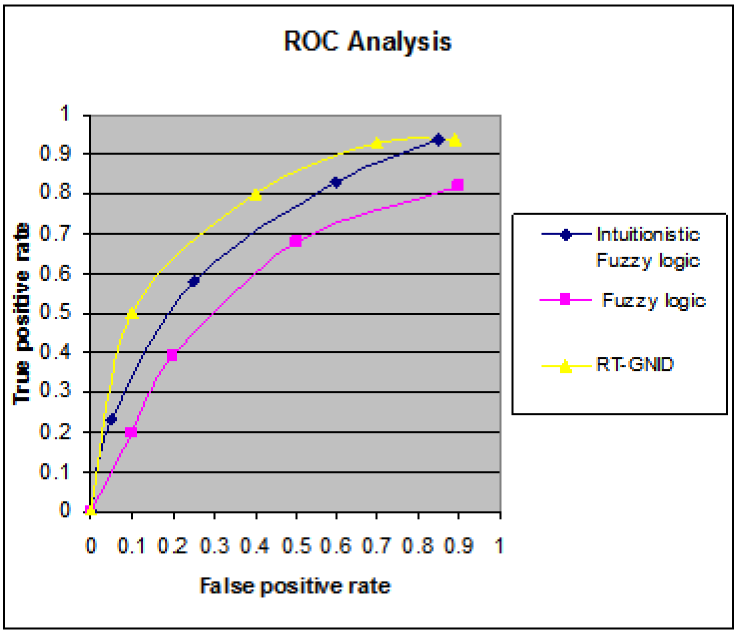


Fig 7 Performance of RT-GNID with other models

According to the Fig. 7, Neutrosophic Logic rule based intrusion detection system outperforms the remaining models fuzzy logic and intuitionistic fuzzy logic. The results shows best lower false alarm rate with a higher detection rate in RT-GNID. Using the degree of membership for truthness, falsity and indeterminacy the performance rate improved significantly. It is feasible to discover the indeterministic rule which needs more consequence when expressing the imprecise examined objects. From the results obtained, it is an evident that the improvised genetic algorithm adapted along with the Neutrosophic logic for this experiment was effectively able to produce a model with the desired characteristics of a high correct detection rate and a low false positive rate from learning over real time environment.

X. CONCLUSION

Employing RT-GNID the proposal of tripartitioning the dataset into normal, abnormal and indeterministic is easily obtained by classifying the dataset in the basis of degree of truthness, falsehood and indeterminacy. This proposed work is the extension of our previous work in which Intuitionistic Fuzzy Logic was implemented. It is observed that the proposed approach catches the imprecision of knowledge, uncertainty due to incomplete knowledge or acquisition errors or stochastic and vagueness due to lack of clear contour or limits can be overcome using the NL based classifier. The primary contribution of this paper is to overcome the problem of incomplete and inconsistent information without danger of trivialization in real-time environment.

REFERENCE

[1]. John E. Canavan , Fundamentals of network security, British Library Cataloguing in Publication Data, ISBN 1-58053-176-8, 2001, ARTECH HOUSE, INC.

[2]. D. E. Denning, “An intrusion detection model,” IEEE Transactions on Software Engineering, vol. 13, no. 2, pp. 222– 232, 1987.

[3]. J. P. Anderson, “Computer security threat monitoring and surveillance,” Tech. Rep., James P. Anderson Co., Fort, Washington, PA, USA, 198

[4]. J. Cannady, “Artificial neural networks for misuse detection,” in Proceedings of the 1998 National Information Systems Security Conference, pp. 443–456, Arlington, VA, USA, 1998.

[5]. A. A. Aburomman and M. B. I. Reaz, “A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems,” Information Sciences, vol. 414, pp. 225–246, 2017.

[6]. A. Tajbakhsh, M. Rahmati, and A. Mirzaei, Intrusion detection using fuzzy association rules, Applied Soft Computing, vol. 9, no. 2, pp. 462–469, 2009.

[7]. T. Ozyer, R. Alhajj, and K. Barker, “Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening,” Journal of Network and Computer Applications, vol. 30, no. 1, pp. 99–113, 2007

[8]. E. Lazcorreta, F. Botella, and A. Fernández-Caballero, “Towards personalized recommendation by two-step modified apriori data mining algorithm,” Expert Systems with Applications, vol. 35, no. 3, pp. 1422–1429, 2008.

[9]. S. Mabu, K. Hirasawa, and J. Hu, “A graph-based evolutionary algorithm: genetic network programming (GNP) and its extension using reinforcement learning,” Evolutionary Computation, vol. 15, no. 3, pp. 369–398, 2007

[10]. F. Smarandache, A unifying field in logics: neutrosophic logic, Multiple-Valued Logic/An International Journal 8 (3) (2002) 385–438, http://www.gallup.unm.edu/_smarandache/eBook-neutrosophics2.pdf.

[11]. J. V. Hansen, P. B. Lowry, R. D. Meservy, and D. M. Mcdonald, “Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection,” Decision Support Systems, vol. 43, no. 4, pp. 1362–1374, 2007.

[12]. F. Smarandache, Neutrosophy, A new branch of philosophy, in multiple-valued logic, An International Journal 8(3) (2002) 297–384.

[13]. F. Smarandache (Eds.), Proceedings of the First International Conference on Neutrosophy, Neutrosophic Logic, Neutrosophic Set, Neutrosophic Probability and Statistics, University of New Mexico, Gallup Campus, Xiquan, Phoenix, 2002, p. 147

[14]. Grigorios N. Beligiannis, Georgios A. Tsirogiannis, Panayotis E. Pintelas,Restartings: a technique to improve classic genetic algorithms’ performance, World Academy of Science, Engineering and Technology, 2005.

[15]. Jonatan Gomez, Dipankar Dasgupta, Olfa Nasraoui, Fabio Gonzalez, Complete Expression Trees for Evolving Fuzzy Classifier Systems with Gentic Algorithms and Application to Network Intrusion Detection, NAFIPS, 2002, 2002 - ieeexplore.ieee.org.