Knowledge-Based Systems 28 (2012) 88–96

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

An ensemble design of intrusion detection system for handling uncertainty using Neutrosophic Logic Classifier

B. Kavitha [a,*], Dr. S. Karthikeyan [b], P. Sheeba Maybell [c]

[a] Research and Development Centre, Bharathiar University, Coimbatore, India,
[b] Department of Information Technology, College of Applied Sciences, Sohar, Oman
[c] Department of Mathematics, Karpagam University, Coimbatore, India

ARTICLE INFO

Article history:
Received 21 July 2011
Received in revised form 1 December 2011
Accepted 5 December 2011
Available online 13 December 2011

Keywords:
Intrusion
KDD cup
Uncertainty
Fuzzy
Neutrosophic
Membership
Neutrosophic
Improvised genetic algorithm

ABSTRACT

In the real world it is a routine that one must deal with uncertainty when security is concerned. Intrusion detection systems offer a new challenge in handling uncertainty due to imprecise knowledge in classifying the normal or abnormal behaviour patterns. In this paper we have introduced an emerging approach for intrusion detection system using Neutrosophic Logic Classifier which is an extension/combination of the fuzzy logic, intuitionistic logic, paraconsistent logic, and the three-valued logics that use an indeterminate value. It is capable of handling fuzzy, vague, incomplete and inconsistent information under one framework. Using this new approach there is an increase in detection rate and the significant decrease in false alarm rate. The proposed method tripartitions the dataset into normal, abnormal and indeterministic based on the degree of membership of truthness, degree of membership of indeterminacy and degree of membership of falsity. The proposed method was tested up on KDD Cup 99 dataset. The Neutrosophic Logic Classifier generates the Neutrosophic rules to determine the intrusion in progress. Improvised genetic algorithm is adopted in order to detect the potential rules for performing better classification. This paper exhibits the efficiency of handling uncertainty in Intrusion detection precisely using Neutrosophic Logic Classifier based Intrusion detection System.

1. Introduction

Intrusion detection (ID) is a kind of security managing scheme for computers and networks. An ID system collects and investigates information from diverse areas within computers or networks to spot potential security violations, which include both intrusion and misuse. ID uses vulnerability assessment, which is a technology developed to assess the security of a computer system or network. ID systems are being developed in response to the ever rising number of attacks on major sites and networks. The safeguarding of security is becoming increasingly difficult, because the possible technologies of attack are becoming ever more sophisticated and at the same time, less technical ability is required for the novice attacker, as proven past methods are easily accessed through the Web.

While the intrusion detection system (IDS) in network is making great progress, it is also facing great challenges [1]. Rule-based systems are most extensively deployed in network intrusion detection products. They are effortless to recognize and use, but necessitate human domain experts to find the rules and their generalization power depends on the expertise knowledge in the attacks. Machine learning and data mining techniques are possible solutions to this drawback, but this heavily depends, again, on the domain experts to tell what features are important to learn [2].

First major challenge in intrusion detection is that we have to identify the veiled intrusions from a huge amount of normal communication activities. Dimensionality reduction is crucial when data mining techniques are applied for intrusion detection. The Data Mining process requires high computational cost when dealing with large data sets [3]. Most of the existing IDS use all 41 features in the network to evaluate and look for intrusive pattern; some of these features are redundant and irrelevant. The drawback of such an approaches leads to time-consuming in detection process and it also degrades the performance of IDS, thus we need to remove the worthless information from the original high dimensional database. To improve the generalization ability, we usually generate a small set of features from the original input variables by feature selection. In our previous work [4] we have applied best first search method to reduce the dimensionality of attributes and result shows 7 potential attributes for classification.

Second Major Challenge is to classify the attack degrees in IDS using data mining. Even if fusion is expected to reduce the variance and improve the detection, there is uncertainty associated with

* Corresponding author.
E-mail address: kavitha_gana2006@yahoo.co.in (B. Kavitha).

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.12.004

every IDS. Uncertainty is an innate feature of intrusion analysis due to the limited views provided by system monitoring tools, IDS and various types of logs [5]. To describe the uncertainty, we should classify the degree of the attack activities, and users can adjust the detection strategy according to the actual situation.

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 behavior 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.

2. Related work

Xinming et al. [6] proposed an empirical approach to the problem of uncertainty where the inferred security implications of low-level observations are captured in a simple logical language augmented with certainty tags. The probabilistic approach [7] for detecting network intrusions using Bayesian networks (BNs) shows that the hand-crafted BN, in general, has outperformed naive Bayesian network and Learned BN. A new evidence model [8] which is an extension and improvement of the classical Dempster–Shafer theory is proposed to improve the probability of detection along with a reduction in the false alarm rate with the proposed fusion algorithm.

Srinivas et al. [9] describes approaches to intrusion detection using neural networks and support vector machines. The key ideas of the research are to discover useful patterns or features that describe user behavior on a system and use the set of relevant features to build classifiers that can recognize anomalies and known Intrusions. The temporal association rules technique generates fuzzy and classical rules [10]. Using short sequences of system calls that running programs perform as discriminators between normal and abnormal operating characteristics [11]. The discriminator uses the Hamming distance as a distance function between short sequences of system calls. If the distance of a particular sequence to the normal sequences is higher than a threshold then the sequence is abnormal.

A novel intrusion detection model [21] adapted artificial immune and mobile agent paradigms for improvising network intrusion detection. Inspired by the theory of artificial immune system a novel model of agents of network danger evaluation [22] is proposed to enhance the self learning ability to adapt continuously varied environments, which provides a good solution for network surveillance. To overcome the problem of handling high dimensional data Rough set theory was tailored [23] for identifying meaningful outliers and analyzing their intentional knowledge for finding the key attribute subset in dataset. A new technique is implemented for overcoming the problem of handling uncertainty problem in intrusion detection system; the method uses the concept of Intuitionistic Fuzzy Logic for classifying the normal and abnormal packets [24].

Many of the research works in the field of IDS try a feasible approach to an improved detection rate. With the increasing traffic and increasing complexity of attacks, there is a high demand for an incredibly high detection rate (usability) and an extremely low false alarm rate (acceptability). Most of the IDSs available in literature show distinct preference available for detecting a certain class of attack with improved accuracy while performing moderately for the other classes of attacks.

This paper describes the central concept underlying the work and a theme that ties together all the arguments in this work.

Table 2
Amount and ratio of data sampling.

Category	Corrected dataset		Randomly selected sampled records	
Normal	60593	19.48%	8883	13.67%
Probe	4166	1.34%	4166	6.4%
DOS	229853	73.9%	35534	54.67%
U2R	70	.02%	70	.11%
R2L	16347	5.26%	16347	25.15%
Total	3,11,029	100%	65000	100%

3. Dataset description

3.1. KDDcup'99 Dataset

In this paper, KDDcup'99 data set is used which is based on the 1998 DARPA [12,13]. Normal connections are created to profile that those expected in a military network and attacks fall into one of the following four categories namely Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R) and Probe. The various types of attack in our experimental dataset which are classified into four categories are shown in the following Table 1

The KDDCup'99 Intrusion Detection benchmark is comprised of 3 components. In this work corrected KDD set is used because a dataset with different statistical distributions than either "10% KDD" or "Whole KDD" is provided by the "Corrected KDD" and is comprised of 14 additional attacks. Hence, the "Corrected KDD" dataset is being used for our experiment. The value of each connection is being predicted by this task.

3.2. Exclusion of dataset

As in our previous work [4] 65000 records have been selected as sample dataset out of 3, 11,029 Corrected KDD dataset connections for the work done by us. However, because the sample number of Probe, U2R, and R2L is being less, the number of records of above attack types will be constant in any sample rate. The remaining records out of 65,000 are 44,417 which are the outcome of excluding the Probe, U2R and R2L types of records. Out of 44417, 20% of Normal connection is selected, and remaining 80% of the dataset is

Table 1
Various attack types.

Categories	Attack types
DoS	Apache2, Back, Land, Mail bomb, Neptune, Pod, process Table, Smurf, Tear drop, Udpstrom
PROBE	IPsweep, Mscan, nMap, Portsweep, Saint, Satan
U2R	Buffer Overflow, http tunnel, load module, perl, root kit, ps, sqlattack, xterm
R2L	Ftpwrite, guesspasswd, imap, multihop, named, phf, send mail, snmp getattack, snmpguess, warezmaster, worm, xlock, xsnoop

accounted by the Dos. The data sampling number and ratio are shown in Table 2.

4. An introduction to neutrosophic logic

4.1. Nonstandard 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 $1^+ = 1 + \varepsilon$, where "1" is its standard part and "ε" its non-standard part, and $^-0 = 0 - \varepsilon$, 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.

4.2. Neutrosophic components

Let T, I, F be standard or non-standard real subsets of the non-standard unit interval $]^-0, 1^+[$, with

$$\sup T = t\_\sup, \quad \inf T = t\_\inf,$$
$$\sup I = i\_\sup, \quad \inf I = i\_\inf,$$
$$\sup F = f\_\sup, \quad \inf F = f\_\inf$$

and

$$n\_\sup = t\_\sup + i\_\sup + f\_\sup,$$
$$n\_\inf = t\_\inf + i\_\inf + f\_\inf.$$

The sets T, I, F are not necessarily intervals, but may be any real sub-unitary subsets, discrete or continuous, single-element, finite, or (countable or uncountable) infinite, union or intersection of various subsets, etc. They may also overlap. The real subsets could represent the relative errors in determining t, i, f (in the case when the subsets T, I, F are reduced to points). Statically T, I, F are subsets. For Example:

The truth value of a proposition may change from a place to another place, for example: the proposition "It is raining" is 0% true, 0% indeterminate and 100% false in Albuquerque (New Mexico), but moving to Las Cruces (New Mexico) the truth value changes and it may be (1,0,0).

4.3. Neutrosophic logic

Smarandache [16] 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 [14,15 and 16]. 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.

A. To maintain the consistency with the classical and fuzzy logics and with probability there is the special case where $T + I + F = 1$.
B. But to refer to intuitionistic logic, which means incomplete information on a variable proposition or event one has $T + I + F < 1$.
C. 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 $n_{\inf} = \inf T + \inf I + \inf F \geqslant {}^-0$, and $n_{\sup} = \sup T + \sup I + \sup F \leqslant 3^+$.

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

5. 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 cannot 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 [19], paraconsistent logic, and the three-valued logics that use an indeterminate value [14,15 and 16].

5.1. 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 $\sup T + \sup F \leqslant 1$.
5. The interval valued intuitionistic fuzzy set, $I = \phi$, $\inf T$, $\sup T$, $\inf F$, $\sup F \in [0,1]$ and $\sup T + \sup F \leqslant 1$.
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 $a \to b$ means that b is a generalization of a.

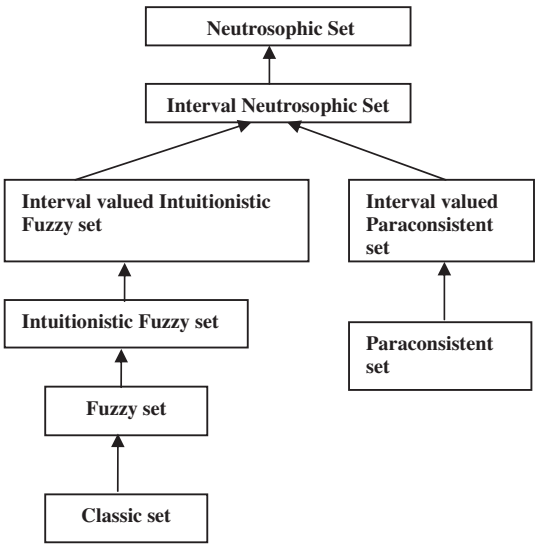

Fig. 1. Relationship among Neutrosophic set and other sets.

5.2. Differences between neutrosophic logic and Intuitionistic Fuzzy Logic

The differences between IFL and NL [20] (and the corresponding intuitionistic fuzzy set and neutrosophic set) are:

- NL can distinguish between absolute truth (truth in all possible worlds, according to Leibniz) and relative truth (truth in at least one world), because NL(absolute truth) = 1^+ while NL(relative truth) = 1. This has application in philosophy. That's why the unitary standard interval [0,1] used in IFL has been extended to the unitary non-standard interval $]^-0,1^+[$ in NL. Similar distinctions for absolute or relative falsehood and absolute or relative indeterminacy are allowed in NL.
- In NL there is no restriction on T, I, F other than they are subsets of $]^-0,1^+[$. Thus: $^-0 \leqslant \inf T + \inf I + \inf F \leqslant \sup T + \sup I + \sup F \leqslant 3+$. This non-restriction allows paraconsistent, dialetheist, and incomplete information to be characterized in NL {i.e. the sum of all three components if they are defined as points, or sum of superior limits of all three components if they are defined as subsets can be >1 (for paraconsistent information coming from different sources) or <1 for incomplete information}, while that information cannot be described in IFL because in IFL the components T (truth), I (indeterminacy), F (falsehood) are restricted either to $t + i + f = 1$ or to $t2 + f2$ [1, if T, I, F are all reduced to the points t, i, f respectively, or to $\sup T + \sup I + -\sup F = 1$ if T, I, F are subsets of [0,1].
- In NL the components T, I, F can also be non-standard subsets included in the unitary non-standard interval $]^-0,1^+[$, not only standard subsets included in the unitary standard interval [0,1] as in IFL.
- NL, like dialetheism, can describe paradoxes, NL (paradox) = (1,I,1), while IFL cannot describe a paradox because the sum of components should be 1 in IFL ([11–13]).
- NL has a better and clear name "Neutrosophic" (which means the neutral part: i.e. neither true nor false), while IFL's name "Intuitionistic" produces confusion with Intuitionistic Logic, which is something different.

6. Proposed model of intrusion detection system using Neutrosophic Logic Classifiers

In this paper we propose a model of an intrusion detection system using NL and improvised genetic algorithm based on data mining techniques. This proposed work is based on the evolutionary design of intrusion detection systems

The proposed approach for intrusion detection system is as follows:

- Step 1: Collecting the dataset from KDD cup 99.
- Step 2: Dataset pre-processing which normalize the dataset.
- Step 3: Applying dimensionality reduction 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 7: After tuning the rules, the testing datasets are validated.

The Fig. 2 depicts the proposed system architecture for classifying the dataset based on NLC.

7. Neutrosophic logic and three class classification

In Fig. 3 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

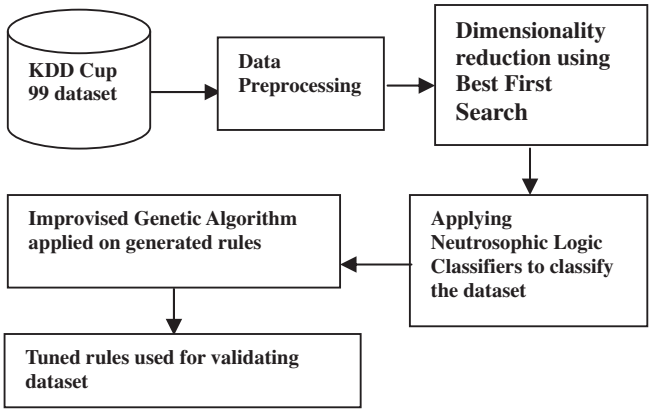

Fig. 2. Proposed system architecture.

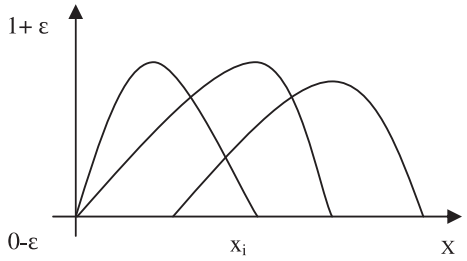

Fig. 3. Membership function of neutrosophic logic.

- 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.

7.1. 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 (protocol type = tcp ^ service = http ^ source_bytes $\leqslant$ 19721 ^ destination_bytes $\leqslant$ 1, 25,015 ^ count $\leqslant$ 326 ^ diffserv errate $\leqslant$ 1 ^ 20 $\leqslant$ destinationhostservercount $\leqslant$ 255) Then
Pattern is normal [0.3]
R_A : If (protocol type = icmp ^ service = ecr_i ^ source_bytes = 1480 ^ destination_bytes = 0 ^ 1 $\leqslant$ count $\leqslant$ 20 ^ diffserverrate = 0 ^ 1 $\leqslant$ destinationhostservercount $\leqslant$ 20) Then
Pattern is abnormal [0.5]
[Pod / DOS]
R_A : If (protocol type = tcp ^ service = private ^ source_bytes = 0 ^ destination_bytes = 0 ^ count $\leqslant$ 2 ^ diffserverrate = 0 ^ destinationhostservercount = 1) Then
Pattern is abnormal [0.5]
[Nmap / Probe]
R_A : If (protocol type = *tcp* ^ service = telnet ^ source_bytes $\leqslant$ 1735 ^ destination_bytes $\leqslant$ 6707 ^ count = 1 ^ diffserverrate = 0 $\wedge$ destinationhostservercount $\leqslant$ 4) Then
Pattern is abnormal [0.5]
[Bufferoverflow/U2R]
R_A : If (protocol type = tcp ^ service = telnet ^ source _bytes = 126 ^ destination_bytes = 179 ^ count $\leqslant$ 3 ^ diffserverrate = 0 ^ destinationhostservercount $\leqslant$ 11) Then
Pattern is abnormal [0.5]
[guess_passwd / R2L]
R_I : If (protocol type = tcp ^ service = private ^ source_bytes = 0 ^ destination_bytes = 0 ^ count = 1 ^ diffserverrate = 0 ^ destinationhostservercount = 1) Then
Pattern is indeterministic [0.2]
R_I : If (protocol type = udp ^ service = private ^ source_bytes = 215 ^ destination_bytes = 0 ^ count = 1 ^ diffserverrate = 0 ^ destinationhostservercount = 1) Then
Pattern is indeterministic [0.2]

The Neutrosophic rule truth value is calculated as the product of the condition truth value by the weight.

$$TV(R) = TV\ (Condition)^{*}Weight$$

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.:

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

where,

N – Represents the Normal class,
A – Represents the Abnormal class and
I – Represents the Indeterministic class

8. Improvised genetic algorithm

Grigorios et.al [17], 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.

8.1. Operator used in genetic algorithm restartings

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

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

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

Table 3
Neutrosophic logic operator.

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

- ***Mutation Operator*** : *Suppose a chromosome* $S_i = \{S_{11}, S_{12}, S_{13} \ldots, S_{1n}\}$ *Select a random integer number* $0 \leqslant r \leqslant n$, *S3 is a mutation of* S_1,
  - $S_3 = \{S_i | if\ i \neq r, S_i \in S1_i, else\ S_i \in random(S_{1i})\}$
- ***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.

9. Emerging 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. [18] 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 (protocol type is .1 AND service is .2 AND source_bytes is HIGH AND destination_bytes is LOW AND count is NOT HIGH AND diff_server_rate is LOW AND destination_host_server_count 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 $class_1 \ldots$
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 *i*th 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. 4 shows the chromosome for a complete rule condition.

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

A – Protocol_type, B- service = C – source_bytes D- destination_bytes E - count F - diff_server_rate G - destination_host_server_count.

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

For the above expression the chromosome representation is shown in the Fig. 5

Consider the following example:

A is .1 AND **B is .2** AND **C is HIGH** AND **D is LOW** 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:

000110011101011011111000110111110 0

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.

9.1. 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

$$TP = \sum_{i=1}^{p} \text{predicted}(\text{normal\_ data}_i),$$

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}	*

Fig. 4. Representation of chromosome for a complete rule condition.

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. 5. coding the expression A is .1 AND B is .2 AND C is HIGH AND D is LOW 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.

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

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

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

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

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

$$\text{Length} = 1 - \frac{\textit{Chromosome length}(\textit{rules})}{10},$$

$$\text{Fitness} = w1^{*}\text{sensitivity} + w2^{*}\ \text{specificity} + w3^{*}\text{length}.$$

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.

10. Experimental result

The effectiveness of handling incomplete and inconsistent information by NL leads to the construction of Emerging Neutrosophic Logic Classifiers for intrusion detection system (ENLCIDS) and tested their performance on the KDD Cup -99 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) i3 2.13 GHz PC with 4 GB RAM. The implementation is done using MATLAB Software.

10.1. Dataset preprocessing

Using uniform distribution algorithm we created a dataset from the original data set with the following property: If the sample number of k patterns is m and the original data set has n samples. Probability to find a sample of class $y = m/n$ samples of the final dataset in 1.0

Each dataset is normalized between 0.0 & 1.0 using the equation.

$$X = \frac{x - \min}{\max - \min},$$

where,

- x – Numerical value,
- min – minimum value for the attribute that x belongs to,
- max – maximum value for the attribute that x belongs.

The non numeric data has the degree of membership value is $^{-}0$ for false and 1^{+} for true.

10.2. Dimensionality reduction

The original dataset is comprised of 41 attributes and one class label. In our previous work we adopted Best First Search method [7] using that we obtained set of reduced dimensionality to 7 potential attributes. The Tables 4 and 5 shows the list of 41 attributes and 7 attributes respectively.

The above said 7 attributes are identified as potential ones to frame the atomic expressions and complete expression tree was developed which eliminates most of the inconsequential rules.

10.3. Implementation of neutrosophic classifier

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.

Table 4
41 ATTRIBURES.

Duration	Protocol type
service	Flag
src_bytes	dst_bytes
land	wrong_fragment
urgent	Hot
num_field_logins	logged_in
num_compromised	root_shell
su_attempted	num_root
num_file_creation	num_shells
num_access_files	num_outbounds_cmds
is_hist_login	is_guest_login
count	srv_count
serror_rate	srv_serror_rate
rerror_rate	srv_rerror_rate
same_srv_rate	diff_srv_rate
srv_diff_host_rate	dst_host_count
dst_host_srv_count	dst_hosdst_same_srv_rate
dst_host_diff_srv_rate	dst_host_same_src_port_rate
dst_host_srv_diff_host_rate	dst_host_serror_rate
dst_host_srv_serror_rate	dst_host_rerror_rate
dst_host_srv_rerror_rate	

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 7.1

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.

10.4. Results and analysis

The false alarm rate and the undetected attack rates are the two factors that define the cost function of an intrusion detection system. The average performance of (ENLCIDS) proposed approach over twenty five test performed is shown in the Table 6.

- FRID – Fuzzy Rule based Intrusion Detection
- IFRID – Intuitionistic Fuzzy Rule based Intrusion Detection
- ENLCRID – Emerging Neutrosophic Logic Classifier Rule based Intrusion Detection

Table 5
7 ATTRIBUTES.

Protocol type	Service
Src_bytes	Dst_bytes
count	diff_srv_rate
dest_host_srv_count.	

Table 6
comparison of the proposed model.

Algorithm	False alarm%	Detection rate%	$o(n)$
RIPPER – Artificial Anomalies	20.0	94.26	878.09
SMART SIFTER	22.3	82.0	465.18
FRID	10.63	95.47	347.19
IFRID	5.03	97.86	305.02
ENLCID	3.19	99.02	258.65

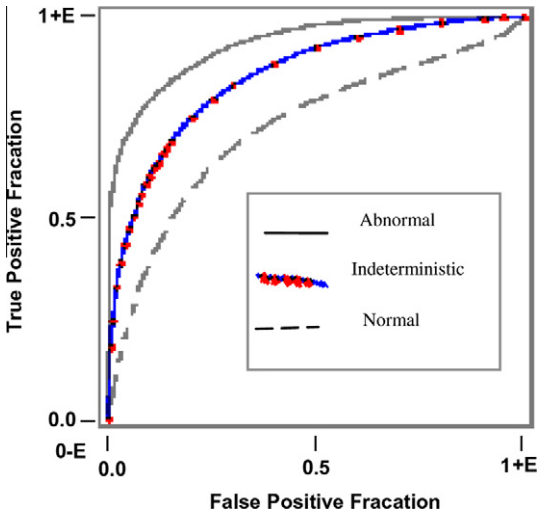

Fig. 6. ROC Curve for Neutrosophic Logic Based Classifier.

Among the five different approaches Emerging Neutrosophic Logic Classifier outperforms. 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^{+}

According to the Fig. 6, Neutrosophic Logic rule for the abnormal class produces the best results (lower false alarm rate with a higher detection rate). Using the degree of membership for normal, abnormal and indeterministic it is possible to identify the indeterministic rule which needs more importance when expressing the imprecise examined objects. From the results obtained, it is evident that the improvised genetic algorithm adapted along with the Intuitionistic Fuzzy logic for this experiment was successfully able to generate a model with the desired characteristics of a high correct detection rate and a low false positive rate from learning over training data set

11. Conclusion

By employing Emerging Neutrosophic Logic Classifiers for intrusion detection system the idea 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.

References

[1] Wei Liu, Research of Data Mining in intrusion detection system and the Uncertainty of the Attack, in: International Symposium on Computer Network and Multimedia Technology, 2009, CNMT 2009, 18–20 January 2009, pp. 1–4, ISBN:978-1-4244-5272-9, INSPEC Accession Number: 11069703.

[2] D. Barbara, S. Jajodia, Applications of Data Mining in Computer Security, Klnwer Academic Publishers, 2002.

[3] Oded Z. Maimon, Lior Rokach, Data Mining and Knowledge Discovery Handbook, Springer, 2005.

[4] B. Kavitha, S. Karthikeyan, B. Chitra, Efficient intrusion detection with reduced dimension using data mining classification methods and their performance comparison, in: V.V. Das et al. (Eds.): BAIP 2010, CCIS 70, 2010, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 96–101.

[5] Xinming Ou, S.R. Rajagopalan, S. Sakthivelmurugan, An empirical approach to modeling uncertainty in intrusion analysis, in: Computer Security Applications Conference, 2009, ACSAC '09, Dec. 2009.

[6] Xinming Ou, S.R. Rajagopalan, S. Sakthivelmurugan, Kansas State Univ., Manhattan, KS, USA, An Empirical Approach to Modeling Uncertainty in Intrusion Analysis, in: Computer Security Applications Conference, 2009, ACSAC '09, pp. 7–11, December 2009.

[7] Kok-Chin Khor, Choo-Yee Ting, S.-P. Amnuaisuk, Fac. of Inf. Technol., Multimedia Univ., Cyberjaya , in: Second Asia International Conference on A Probabilistic Approach for Network Intrusion Detection Modeling & Simulation, AICMS 08, 13–15 May 2008.

[8] Ciza Thomas, N. Balakrishnan, Modified Evidence Theory for Performance Enhancement of Intrusion Detection Systems, SERC, Indian Institute of Science, India.

[9] Srinivas Mukkamala, Guadalupe Janoski, Andrew Sung, Intrusion detection using neural networks and support vector machines, in: Proceedings of IEEE Intrnational Joint Conference on Neural Networks, 2002, pp. 1702–1701.

[10] Yingjiu Li et al., Enhancing profiles for anomaly detection using time granularities, Center for secure information systems, Journal of Computer Security, in press.

[11] Susan Bridges, Rayford Vaughn, Fuzzy data mining and genetic algorithms applied to intrusion detection, in: Proceedings twenty third National Information Security Conference, October 1–19, 2000.

[12] Stefan Axelsson, Intrusion detection systems: a survey and taxonomy, Technical Report No 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.

[13] KDDCup99datasets,The UCI KDD Archive : http:/ kdd.ics.ucs.edu / databases / kddcup99 /kddcup99.html.

[14] 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.

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

[16] 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, www.gallup.unm.edu/~smarandache/NeutrosophicProceedings.pdf.

[17] 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.

[18] 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.

[19] K. Atanassov, Intuitionistic fuzzy sets, Fuzzy Sets and Systems 20 (1986) 87–96.

[20] Charles Ashbacher, Introduction To Neutrosophic Logic, American Research Press, Rehoboth 2002.

[21] Azzedine Boukerche, Renato B. Machado, Kathia R.L. Jucá, João Bosco M. Sobral, Mirela S.M.A. Notare, An agent based and biological inspired real-time intrusion detection and security model for computer network operations, Computer Communications 30 (13) (2007) 2649–2660.

[22] Jin Yang, XiaoJie Liu, Tao Li, Gang Liang, SunJun Liu, Distributed agents model for intrusion detection based on AIS, Knowledge-Based Systems 22 (2) (2009) 115–119.

[23] Peng Yang, Qingsheng Zhu, Finding key attribute subset in dataset for outlier detection, Knowledge-Based Systems 24 (2) (2011) 269–274.

[24] B. Kavitha, S. Karthikeyan, P. Sheeba Maybell, Emerging intuitionistic fuzzy classifiers for intrusion detection system, Journal of Advances in Information Technology 2 (2) (2011).