ISSN:0975-9646

Ant Colony Optimization Based Hyper Pipes Classifier for IDS

Kumari Babita and Piyush Singh

Department of Computer Engineering,
RKDF IST(Bhopal), Bhopal , INDIA

***Abstract*— With the dramatically development of internet, Security of network traffic is becoming a major issue of computer network system. Attacks on the network are increasing day-by-day. The most publicized attack on network traffic is considered as Intrusion. Intrusion detection system has been used for ascertaining intrusion and to preserve the security goals of information from attacks. Data mining techniques are used to monitor and analyze large amount of network data & classify these network data into anomalous and normal data. Since data comes from various sources, network traffic is large. Data mining techniques such as classification and clustering are applied to build Intrusion detection system. An effective Intrusion detection system requires high detection rate, low false alarm rate as well as high accuracy. This paper presents the review on IDS and different Data mining techniques applied on IDS for the effective detection of pattern for both malicious and normal activities in network, which helps to develop secure information system. This article tells about the new method of IDS with Ant colony optimization and Hyper pipes classifier.**

Keywords: Intrusion Detection, Ant Colony Optimization, Hyper Pipe Classifier, Sensitivity, PPV.

INTRODUCTION

An intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource. Intrusion detection is classified into two types: misuse intrusion detection and anomaly intrusion detection. Misuse intrusion detection uses well-defined patterns of the attack that exploit weaknesses in system and application software to identify the intrusions. These patterns are encoded in advance and used to match against the user behavior to detect intrusion. Anomaly intrusion detection uses the normal usage behavior patterns to identify the intrusion. The normal usage patterns are constructed from the statistical measures of the system features. The behavior of the user is observed and any deviation from the constructed normal behavior is detected as an intrusion (Denning, 1987; Summers, 1997). In Distributed Intrusion Detection System (DIDS) conventional intrusion detection system are embedded inside intelligent agents and are deployed over a large network. In a distributed environment, IDS agents communicate with each other, or with a central server. By having these co-operative agents distributed across a network, incident analysts, network operations, and security personnel are able to get a broader view of what is occurring on their network as a whole. Distributed monitoring allows early detection of planned and coordinated attacks, thereby allowing network administrators to take preventive measures. DIDS also helps to control the spreading of worms, improves network monitoring and incident analysis, attack tracing and so on. It also helps to detect new threats from unauthorized users, back-door attackers and hackers to the network across multiple locations, which are geographically separated (Abraham and Thomas, 2005). In a DIDS it is important to ensure that the individual IDS are lightweight and accurate [1].

Data mining approaches for intrusion detection were first implemented in mining audit data for automated models for intrusion detection (Barbara et al., 2001;Cohen, 1996; Lee et al., 1999). Several data mining algorithms are applied to audit data to compute models that accurately capture the actual behavior of intrusions as well as normal activities. Audit data analysis and mining combine the association rules and classification algorithm to discover attacks in audit data. Soft Computing (SC) is an innovative approach to construct computationally intelligent systems consisting of artificial neural networks, fuzzy inference systems, approximate and derivative free optimization methods such as evolutionary computation, etc. (Zadeh, 1998). This paper introduces three fuzzy rule-based classifiers (Abraham et al., 2004) and compares its performance with Linear Genetic Programming (LGP) (Abraham, 2004), Support Vector Machines (SVM) (Vapnik, 1995) and Decision Trees (DT) (Brieman et al., 1984; Peddabachigari et al., 2004). Further, we modeled Soft Computing (SC)-based IDS (SCIDS) (Abraham et al.,2004) as a combination of different classifiers to model lightweight and more accurate (heavy weight) IDS. The rest of the paper is organized as follows. Section 2 provides a brief overview of the research on distributed intrusion detection systems. Soft computing for intrusion detection is introduced in Section 3 followed by the importance of attribute reduction. [2]

INTRUSION DETECTION SYSTEM

The concept of IDS was proposed by Denning(1987), to identify, detect and trace the intrusion[3]. An IDS is a combination of software and hardware which are used for detecting intrusion [4]. It gathers and analyzes the network traffic & detects the malicious patterns and finally alert to the proper authority. The main function of IDS includes:[5]

- Monitoring and analyzing the information gathered from both user and system activities.
- Analyzing configurations of system and evaluating the file integrity and system integrity.
- For static records, it finds out the abnormal pattern.
- To recognize abnormal pattern, it use static records and alert to system administrator.

A. Classification of IDS

According to techniques used for intrusion detection based on whether attack's patterns are known or unknown, IDS classified into two category [6][7]:

(1) Misuse detection

(2) Anomaly detection

Misuse detection: It is Signature based IDS where detection of intrusion is based on the behaviors of known attacks like antivirus software. Antivirus software compares the data with known code of virus. In Misuse detection, pattern of known malicious activity is stored in the dataset and identify suspicious data by comparing new instances with the stored pattern of attacks.

Anomaly detection: [5][8] It is different from Misuse detection. Here baseline of normal data in network data in network eg load on network traffic, protocol and packet size etc is defined by system administrator and according to this baseline, Anomaly detector monitors new instances. The new instances are compared with the baseline, if there is any deviation from baseline, data is notified as intrusion. For this reason, it is also called behavior based Intrusion detection system.

B. Working of Intrusion Detection System

Author presents 4-steps for working of IDS.

1) Data Acquisition: Data is collected from various sources by using particular software.

2) Feature Selection: Huge amount of data is collected from network traffic. So dataset for IDS becomes large. For working on large dataset generate feature vectors, which contains only necessary data.

3) Analysis: In this step, Collected data is analyzed to determine whether data is suspicious or not. Here, various Data mining techniques are used for Intrusion detection.

4) Action: IDS alarms the administrator about attack which has been detected.

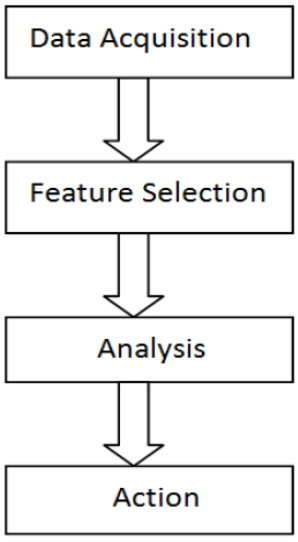

Fig. 1. Working of Intrusion Detection System

Normalcy Classifier and a Hybrid Intrusion Detection System

A current network IDS setup is shown in Figure 1. Adding in a front-end with the capability to replicate the detections of a signature NIDS creates a hybrid system that can significantly improve the speed at equivalent false alarm rates but with a slightly higher false negative rate. For a hybrid system, the labeling and analysis of detection does not need to be implemented because a version of the signature NIDS should be run on the detections for labeling and analysis of the suspicious network traffic as shown in Figure 2. This hybrid system will outperform the signature NIDS as a standalone in speed since the high percentage of network traffic will be classified as normal and not sent to the labeler. It will also be more scalable, since additional normalcy classifiers can be run with significantly less overhead. The resulting system will produce the same level of labeling quality since the abnormal traffic would be routed through the signature component. The level of false alarms would not rise since the signature component would be run on the abnormal output from the normalcy classifier to reduce false alarms. The hybrid system would run faster, scale more easily, and use far less resources than a series of signature NIDS instances. The cost is the slightly increased false negative rate caused by missed detections in the normalcy classifier. However, the abnormal output from the normalcy classifier may contain significant information about a new or unrecognized attack pattern. This output can be sent to an analyst or to an anomaly classifier.

Fuzzy Rule Based Systems

Fuzzy logic has proved to be a powerful tool for decision making to handle and manipulate imprecise and noisy data. The notion central to fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0.0, 1.0], with 0.0 representing absolute falseness and 1.0 representing absolute truth. A fuzzy system is characterized by a set of linguistic statements based on expert knowledge. The expert knowledge is usually in the form of if-then rules.

Definition 1: Let X be some set of objects, with elements noted as x. Thus,

$X = \{x\}$.

Definition 2: A fuzzy set A in X is characterized by a membership function which are easily implemented by fuzzy conditional statements. In the case of fuzzy statement if the antecedent is true to some degree of membership then the consequent is also true to that same degree.

A simple rule structure: If antecedent then consequent

A simple rule: If variable1 is low and variable2 is high then output is benign else output is malignant

In a fuzzy classification system, a case or an object can be classified by applying a set of fuzzy rules based on the linguistic values of its attributes. Every rule has a weight, which is a number between 0 and 1 and this is applied to the number given by the antecedent. It involves 2 distinct parts. First the antecedent is evaluated, which in turn involves fuzzifying the input and applying any necessary fuzzy operators and second applying that result to the consequent known as inference. To build a fuzzy

classification system, the most difficult task is to find a set of fuzzy rules pertaining to the specific classification problem.

We explored three fuzzy rule generation methods for intrusion detection systems. Let us assume that we have a n dimensional c-class pattern classification problem whose pattern space is an n-dimensional unit cube [0, 1]n. We also assume that m patterns xp = (xpl,...,xpn) , p =1,2,...,m, are given for generating fuzzy if-then rules where xp □ [0,1] for p =1,2,..., m, i =1,2,...,n where xp □ [0,1] for p =1,2,..., m, i =1,2,...,n.

Linear Genetic Programming (LGP)

Linear genetic programming is a variant of the GP technique that acts on linear genomes [9]. Its main characteristics in comparison to tree-based GP are that the evolvable units are not expressions of a functional programming language (like LISP), but the programs of an imperative language (like c/c ++). An alternate approach is to evolve a computer program at the machine code level, using lower level representations for the individuals. This can tremendously hasten the evolution process as, no matter how an individual is initially represented, finally it always has to be represented as a piece of machine code, as fitness evaluation requires physical execution of the individuals.

The basic unit of evolution here is a native machine code instruction that runs on the floating-point processor unit (FPU). Since different instructions may have different sizes, here instructions are clubbed up together to form instruction blocks of 32 bits each. The instruction blocks hold one or more native machine code instructions, depending on the sizes of the instructions. A crossover point can occur only between instructions and is prohibited from occurring within an instruction. However the mutation operation does not have any such restriction. In this research a steady state genetic programming approach was used to manage the memory more effectively [10].

Neural Learning of Fuzzy Rules (FR3)

The derivation of if-then rules and corresponding membership functions depends heavily on the a priori knowledge about the system under consideration. However there is no systematic way to transform experiences of knowledge of human experts to the knowledge base of a Fuzzy Inference System (FIS). In a fused neuro-fuzzy architecture, neural network learning algorithms are used to determine the parameters of fuzzy inference system (membership functions and number of rules). Fused neuro-fuzzy systems share data structures and knowledge representations. A common way to apply a learning algorithm to a fuzzy system is to represent it in a special neural network-like architecture. An Evolving Fuzzy Neural Network (EFuNN) implements a Mamdani type FIS and all nodes are created during learning. The nodes representing membership functions (MF) can be modified during learning. Each input variable is represented here by a group of spatially arranged neurons to represent a fuzzy quantization of this variable. New neurons can evolve in this layer if, for a given input vector, the corresponding variable value does not belong to any of the existing MF to a degree greater than a membership threshold. Technical details of the learning algorithm are given in [11].

III. VARIOUS CLASSIFIERS

(A) TREE BASED CLASSSIFIERS

Decision tree is one of the most popular and efficient technique in data mining which is established and well-explored by many researchers. Decision trees are categorized as a supervised method that trying to find the relationship between input attributes and target attributes which represent the relationship in structure as a model. The model constructed by using input attributes to predict target However, some decision tree algorithms may produce a large structure of tree size and it is difficult to understand. J48 is an implementation of C4.5 algorithm [Witten & Frank, 2005]. C4.5 was a version earlier algorithm developed by J. Ross Quinlan. There two methods in pruning support by J48 first are known as subtree replacement, it work by replacing nodes in decision tree with leaf [Floriana Esposito et al., 1997; Mohamed et al., 2012; Mandal et al., 2013]. Basically by reduce the number of test with certain path. It works with the process of starting from leaves that overall formed tree and do a backward toward the root. The second type implemented in J48 is subtree raising by moved nodes upwards toward the root of tree and also replacing other nodes on the same way. According to Zhao and Zhang (2007), C4.5 algorithm produce decision tree classification for a given dataset by recursive division of the data and the decision tree is grown using Depth-first strategy. On data testing this algorithm will emphasized on splitting dataset and by selecting a test that will give best result in information gain. In discrete attributes as well, these algorithms consider a test with a result of many as the number of different values and test binary attribute for each attribute will continue to grow in different values each attribute will be considered. In order to gather the entropy gain of all these binary tests efficiently, the training data set belonging to the node in consideration is sorted for the values of the continuous attribute and the entropy gains of the binary cut based on each distinct values are calculated in one scan of the sorted data. This process is repeated for each continuous attributes [9].

A classification technique (or classifier) is a systematic approach to building classification models from an input data set. Examples include decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and native Bayes classifiers. Each technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before. Therefore, a key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records.

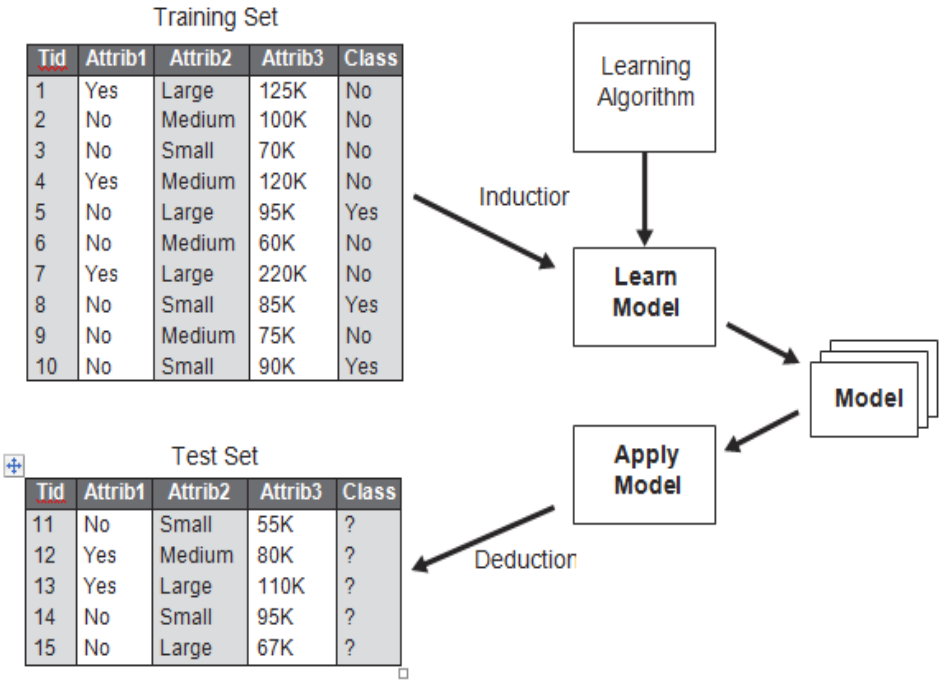

Figure 4.3. General approach for building a classification model.

Figure 4.3 shows a general approach for solving classification problems. First, a training set consisting of records whose class labels are known must be provided. The training set is used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels. Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table known as a confusion matrix. This depicts the confusion matrix for a binary classification problem. Each entry fij in this table denotes the number of records from class i predicted to be of class j. For instance, f01 is the number of records from class 0 incorrectly predicted as class 1. Based on the entries in the confusion matrix, the total number of correct predictions made by the model is (f11 + f00) and the total number of incorrect predictions is (f10 + f01).

Although a confusion matrix provides the information needed to determine how well a classification model performs, summarizing this information with a single number would make it more convenient to compare the performance of different models [4-12].

(B) SOFT COMPUTING BASE

Soft Computing (SC) is an innovative approach to construct computationally intelligent systems consisting of artificial neural networks, fuzzy inference systems, approximate reasoning and derivative free optimization methods such as evolutionary computation etc. In contrast with conventional artificial intelligence techniques which only deal with precision, certainty and rigor the guiding principle of soft computing is to exploit the tolerance for imprecision, uncertainty, low solution cost, robustness, partial truth to achieve tractability and better rapport with reality [9].

(C) OTHERS.

A multiple classifier system approach

In the previous section, we have pointed out that three types of features can be extracted from network traffic data. Each feature category provides information that can be used to discriminate between attacks and normal traffic. In particular, when an attack is performed against a computer network, a "signature" related to that attack may be found in each feature category. For each attack type, network analysts try to design effective attack "signatures" by selecting the more effective subsets of features according to their experience and intuition. On the other hand, pattern recognition tools have been designed to process the entire available feature set to extract more effective signatures than the ones hand-coded by network analysts.

A pattern recognition approach based on the multiple classifiers paradigm can further exploit the above experimental observation that attack evidence can be collected separately in different feature subspaces. First each feature subspace is used independently to perform attack detection. Then the evidence is combined to produce the final decision. This process reflects the behaviour of network security experts, who usually look at different traffic statistics in order to produce reliable attack signatures, i.e., signatures providing effective attack detection and a very low false alarm rate. In addition, the generalization capabilities of pattern recognition algorithms

can allow for the detection of novel attacks that signatures designed by human experts usually do not detect [13].

LITERATURE REVIEW

We developed a hybrid design to a NIDS that enables the seamless insertion of a machine learning component into a signature NIDS system that significantly improves throughput as well as captures additional networking traffic that is similar to known attack traffic. The throughput improvement by incorporating a normalcy classifier is significant, estimated to be the inverse of the false alarm rate which can easily net a factor of 1000. However, this can be diminished by updates that can trigger a retraining of the normalcy classifier. The addition of a normalcy classifier front-end also makes the system more highly scalable and distributable than the signature-based NIDS. The new hybrid design also allows distributed updates and retraining of the normalcy classifier to stay up-to-date with current threats, and makes a number of important performance and quality guarantees. The distributable hybrid implementation is very useful for securing wireless networks with multiple access points.

This system design also has the capability to recognize new attacks that are similar to known attack signatures. The hybrid design also can provide significant information on new attack traffic. By finding the signature of suspicious traffic that is similar to the signature of a known attack, it can be isolated and analyzed as a potential variant of a known attack.[14]

With rapid growth of computer networks during the past few years, network security has become a crucial issue. Among the various network security measures, intrusion detection systems (IDS) play a vital role to integrity, confidentiality and availability of resources. It seems that the presence of uncertainty and the imprecise nature of the intrusions make fuzzy systems suitable for such systems. Fuzzy systems are not normally adaptive and have not the ability to construct models solely based on the target system's sample data. One of the successful approaches which are incorporated fuzzy systems with adaptation and learning capabilities is the neural fuzzy method. The main objective of this work is to utilize ANFIS (Adaptive Neuro Fuzzy Inference System) as a classifier to detect intrusions in computer networks. This paper evaluates performance of ANFIS in the forms of binary and multi-classifier to categorize activities of a system into normal and suspicious or intrusive activities. Experiments for evaluation of the classifiers were performed with the KDD Cup 99 intrusion detection dataset. The Overall Results show that ANFIS can be effective in detecting various intrusions [15] [26].

PROPOSED WORK

This section is dedicated to the flowchart and algorithm of the proposed work.

1. Load NID Dataset
1. Scan and capture NID n*m-1.
2. Scan and capture labels and associate with records
3. Optimize NID with **Ant Colony Optimization**
    a. Initialise pheromone parameters
    b. Generate initial populations (ants)
    c. For each individual ant calculate fitness
    d. For each ant determine its best position
    e. If best global ant determined
    f. Update the pheromone trail
    g. Else
    h. Go to ‘b’
    i. Output is optimized values
4. **Hyper pipes classifier** train by NID train
5. **Hyper Pipes classifier** NID test
6. Output is classified NID
7. Observe classifier ability

Figure 2: Proposed Algorithm

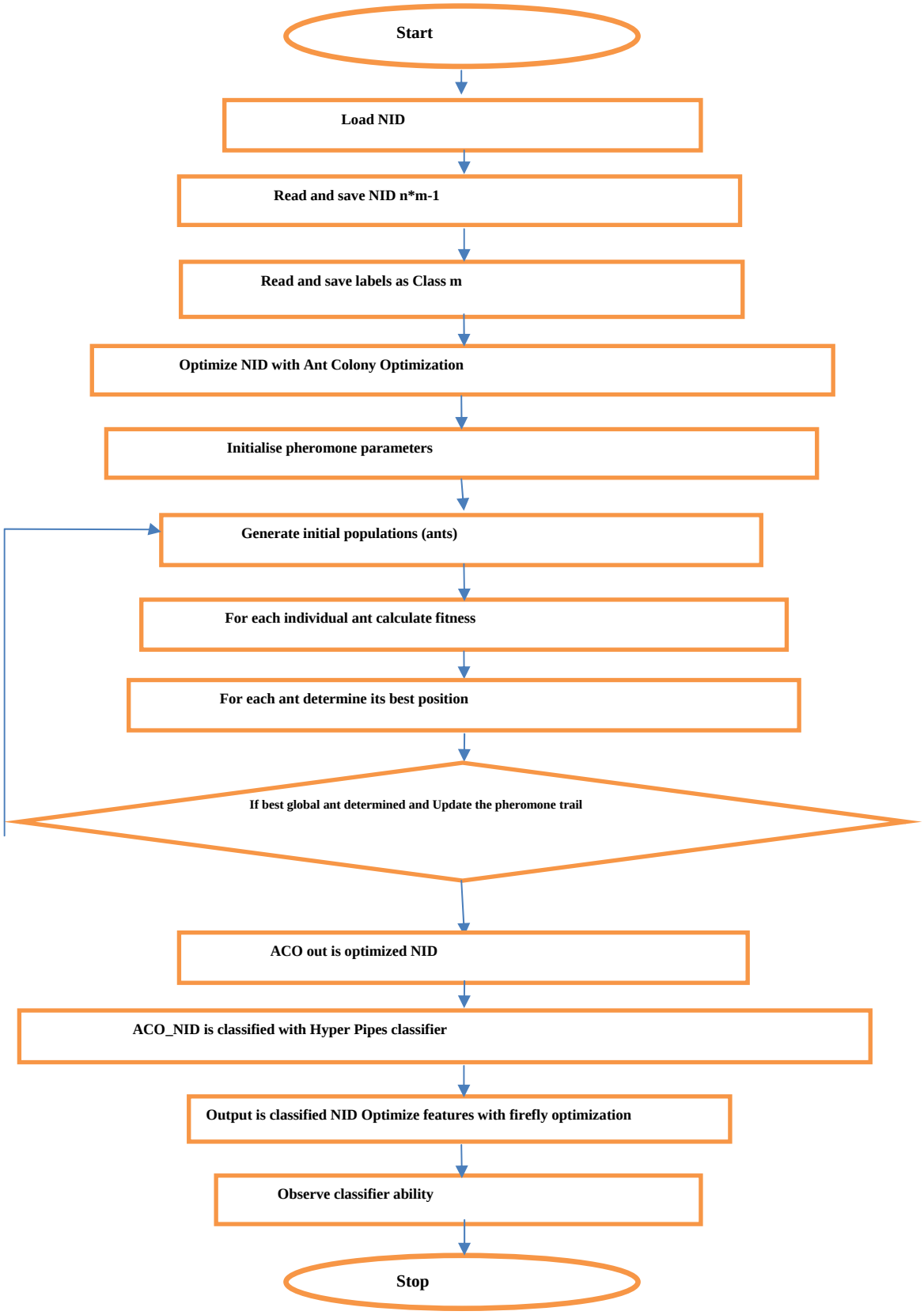

Figure 3: Proposed Architecture

RESULT ANALYSIS

This section is separated into three various sub-parts:

1. **Dataset:** This is shown in table I.

Table I: Dataset

Dataset Detail	Name	Type
	KYOTO 2006+	Network Intrusion
File Detail	Name	Number of Instances
	20070117	61744

Detail of various attacks and normal records in the dataset: This is shown in table II.

Table II: Detail of dataset

Attacks Types	Class	Number of Attributes	
Unknown Attacks	-2	Have	Used
Known Attacks	-1	Conventional(14)	14
Normal	1	Additional(10)	Labels(1)

2. **System Configuration**: IT is shown in table III.

TABLE III : System details

Model:	Sony Vaio
Processor:	Intel® Core™ I5-2450M 2.5GHz
RAM:	4GB
System Type:	64 Bit Operating System
Windows Edition:	Windows 10 Home
Matlab	R2014a

3. **Results**

Positive Predictive Value: It is static value which shows the fraction of retrieved instances that are relevant. It is shown in Table IV and Figure 4.

Table IV: PPV of Existing and proposed work

	Existing	Proposed
PPV (Positive Predictive Value)	0.8925	0.9715

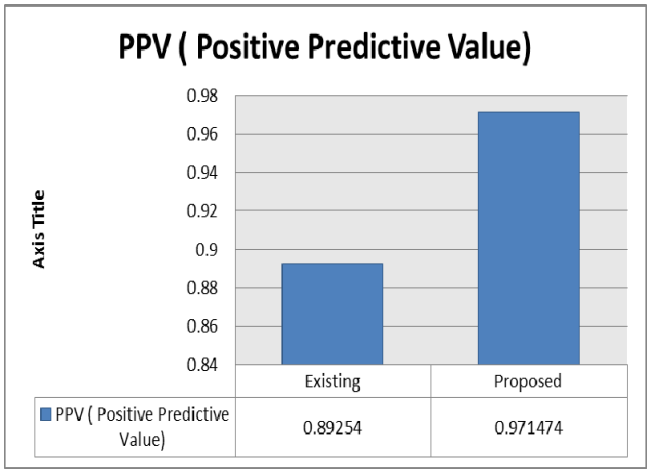

Figure 4: PPV of Existing and proposed work

Sensitivity: It is static value which shows the quantifies the avoiding of false negatives, as specificity does for false positives. It is shown in Table V and Figure 5.

Table V: Sensitivity of Existing and proposed work

	Existing	Proposed
Sensitivity	0.5615	0.9997

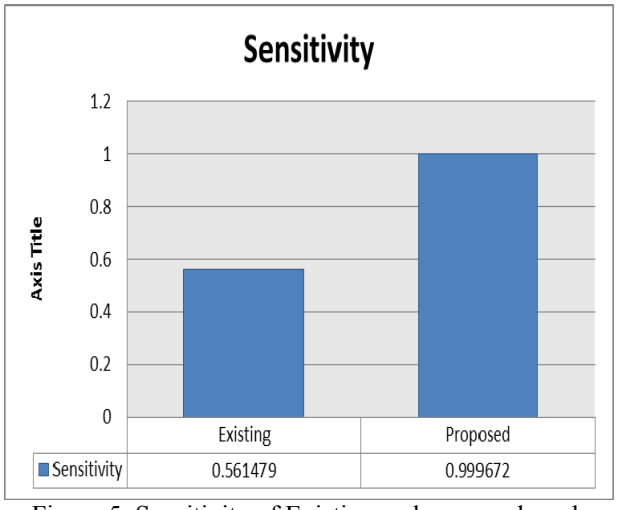

Figure 5: Sensitivity of Existing and proposed work

CONCLUSION

This article is all about the new proposal of the method against Intrusions. In this article the Intrusion detection is done. This intrusion detection is done with the help of Ant colony optimization along with the Table Hyper Pipe Classifier. The figure 4 and 5 along with the Table 4 and 5, clearly show that the performance of the proposed work is

$$= \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

far improved on existing work.

REFERENCES

[1] International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 6, June 2014, "Data Mining Techniques for Intrusion Detection: A Review".

[2] Journal of Network and Computer Applications 30 (2007) 81–98, "D-SCIDS: Distributed soft computing intrusion detection system".

[3] P Amudha and H Abdul Rauf, "Performance Analysis of Data Mining Approaches in Intrusion Detection", IEEE, 2011

[4] Deepthy K Denatious & Anita John, "Survey on Data Mining Techniques to Enhance Intrusion Detection", International Conference on Computer Communication and Informatics (ICCCI - 2012), Jan. 10 – 12, 2012, Coimbatore, INDIA

[5] David Ndumiyana, Richard Gotora and Hilton Chikwiriro, "Data Mining Techniques in Intrusion Detection: Tightening Network Security", International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 5, May – 2013

[6] Rung-Ching Chen, Kai-Fan Cheng and Chia-Fen Hsieh, "Using Rough Set And Support Vector Machine For Network Intrusion Detection", International Journal of Network Security & Its Applications (IJNSA), Vol 1, No 1, April 2009.

[7] Muhammad K. Asif, Talha A. Khan,Talha A. Taj, Umar Naeem and Sufyan Yakoob, " Network Intrusion Detection and its Strategic Importance", Business Engineering and Industrial Applications Colloquium(BEIAC), IEEE, 2013.

[8] Kapil Wankhade, Sadia Patka and Ravindra Thools, "An Efficient Approach for Intrusion Detection Using Data Mining Methods", IEEE 2013

[9] The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Vol. 2, No. 3, May 2014, "A Comparative Study on Disease Classification using Different Soft Computing Techniques".

[10] Abraham A., Evolutionary Computation in Intelligent Web Management, Evolutionary Computing in Data Mining, Ghosh A. and Jain L.C. (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag Germany, 2004.

[11] Kasabov N., Evolving Fuzzy Neural Networks - Algorithms, Applications and Biological Motivation, in Yamakawa T and Matsumoto G (Eds), Methodologies for the Conception, Design and Application of Soft Computing, World Scientific, pp. 271-274, 1998.

[12] "Classification: Basic Concepts, Decision Trees, and Model Evaluation" 2012

[13] Giorgio Giacinto,Fabio Roli, and Luca Didaci, Department of Electrical and Electronic Engineering – University of Cagliari, Italy, "Fusion of Multiple Classifiers for Intrusion Detection in Computer Networks".

[14] David Tahmoush,University of Maryland, University College, Maryland, USA, "A Distributable Hybrid Intrusion Detection System for Securing Wireless Networks".

[15] Conference Paper · July 2006, DOI: 10.1109/ICOCI.2006.5276608. Source: IEEE Xplore "Network intrusion detection based on Neuro-Fuzzy classification"

[16] Zadeh L. A., Roles of Soft Computing and Fuzzy Logic in the Conception, Design and Deployment of Information/Intelligent Systems, Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, O. Kaynak, L.A. Zadeh, B. Turksen, I.J. Rudas (Eds.), pp 1-9, 1998.