

## Review of Computer Engineering Research

2022 Vol. 9, No. 2, pp. 109-121.

ISSN(e): 2410-9142




ISSN(p): 2412-4281

DOI: 10.18488/76.v9i2.3082

© 2022 Conscientia Beam. All Rights Reserved.



# PERFORMANCE EVALUATION OF NETWORK INTRUSION DETECTION SYSTEM FOR DETECTING ZERO-DAY ATTACKS: SNORT-XSS ALGORITHM

 **Srinivas Mishra**<sup>1+</sup>  
 **Sateesh Kumar Pradhan**<sup>2</sup>  
 **Subhendu Kumar Rath**<sup>3</sup>

<sup>1,2</sup>Biju Patnaik University of Technology, Rourkela, Odisha, India.

<sup>1</sup>Email: [srinivas\\_mishra@yahoo.com](mailto:srinivas_mishra@yahoo.com) Tel: +91-9348361302

<sup>2</sup>Email: [rath.subhendu@gmail.com](mailto:rath.subhendu@gmail.com) Tel: +91-9861245967

<sup>3</sup>Utkal University, Vani Vihar, Bhubaneswar, Odisha, India.

<sup>3</sup>Email: [sateesh1960@gmail.com](mailto:sateesh1960@gmail.com) Tel: +91-9437001231



(+ Corresponding author)

## ABSTRACT

### Article History

Received: 14 March 2022

Revised: 27 June 2022

Accepted: 12 July 2022

Published: 4 August 2022

### Keywords

Snort-XSS

Intrusion detection

Soft computing

Fuzzy logic

Artificial neural network

Zero-day attack

Novel attack

KDD cup'99 dataset.

The main objective of Intrusion Detection and Prevention Systems is to provide a method of detecting and preventing malicious behaviors in a network system to minimize the harm caused by attackers. In this article, a survey of the techniques applied for the identification and classification of attacks based on KDD Cup'99 and DARPA data set is discussed, and from the open issues a new and a proficient method called SNORT-XSS algorithm is anticipated and implemented that can recognize and classify real time intrusions including zero day attacks. For this research, the SNORT open source tool developed by CISCO Systems was used to describe rules from the existing data collected from DARPA and KDD Cup'99 dataset. Fuzzy Reasoning system is applied to organize the rules into fuzzy sets that reduces true negative and false positive rate. The advantage of Feed Forward Neural Network with Back Propagation of Errors from Artificial Neuron Networks is considered for training, validating and testing the proposed system. The experimental results achieved by preprocessing anomalous behaviors in a network and the detection rate of zero-day attacks or novel attacks were very promising and were beyond expectations. The precision values of the proposed model were 98.93% and 98.89% respectively, and detection rate of Probe and DoS attacks were greater than 98%. The false positive and true negative rate is almost negligible. It was noticed that the best categorization was acquired at epoch numbers from 50 to 55 with a mean squared error of 0.004.

**Contribution/Originality:** This method combines the advantages of Fuzzy Reasoning and Artificial Neural Network techniques for training, validating and testing the proposed system to minimize true negative and false positive rate during detecting and preventing intrusions in a network.

## 1. INTRODUCTION

The rapid growth of interconnected networks that includes internet and cloud infrastructures are the main platform for cyber attackers to attack, hack and destroy any machine or system. The attackers mainly focus on breaching defense, BFSI (Banking and Financial Services Institutions) sector and personal data for their sole benefit[1]. At present, recognition and prevention of infringements in the field of computers and network security has become a leading area of research in computer science and consequently several revolutionary techniques have been proposed and applied to these schemes. The information revolution has matured in recent years, the possibilities and viewpoints are limitless; unfortunately, the risk and chances of mean attacks also increased rapidly. To overcome this, intrusion detection systems provide a defense tool against intrusions. Network Intrusion

Detection and Prevention System (NIDPS) is an arrangement of software and hardware tools that facilitates identification of user characteristics over the computer network [2]. Pattern matching is the commonly used misuse detection practice. For example, SNORT is an eminent open source and signature based NIDS tool apart from other available misuse detection tools similar to policy based schemes [3].

### 1.1. Zero-Day Attack

A well trending topic of discussion in these days is the “Zero Day Attack”. Researchers have exhausted naming this incident in different ways like “zero hour attack”, “novel attack”, and “unknown attack” which can be defined as the attacks for which no patch or signature is available in the IDPS’s repository [4]. The objective behind this research is to develop a system that can detect and prevent novel attacks along with producing signatures at the time of an unknown attack incidence. The widely accepted method of protecting networks from malware is the anti-malware solutions that have pre-defined signatures. These anti-malware solutions are based on the technique of blacklisting. However, this method is useless against the well-planned malware attacks on the zero-day. At the outset, the research of zero-day threats and its importance was initiated from the identification and signature assignment methods [5]. However, the hit ratio, accuracy and success of the detection technique is lower than that of the failure rates. In reality, detecting the victims and assigning signatures to them alone is not enough, as the recent trends show that the reality is quite opposite.

### 1.2. KDD Cup’99 and DARPA Dataset

In this research, standard KDD Cup’99 and DARPA dataset are used for training, validating and testing the proposed system. The KDD’99 dataset family revealed by the University of California [6], consisting of “The DARPA dataset”, “The KDD Cup’99 dataset” and “NSL-KDD dataset” is considered for this research. Table 1 presents the dataset consisting of a variety of attacks like Denial of Service (DoS), Root to Local (R2L), User to Root (U2R) and Probing attacks [7].

Table 1. KDD’99 dataset family.

KDD Dataset Family	Authentic Records	Distinct Records	Rejection rate
Malicious	3,814,542	251,065	93.1%
Standard	961,670	801,706	18.43%
Total	4,776,212	1,052,771	75.15%

### 1.3. SNORT-XSS

At present SNORT, developed by CISCO Systems, is the widely used, free, lightweight and open source tool for identifying intrusions in a computer network system because of its simplicity in writing rule sets and easy to implement and organize [8]. Without any effort, one can flexibly design and incorporate the necessary rules to the rules database, because the rules are written in shared object files (.so) that can be easily edited and can be maintained using a simple notepad. Meanwhile at the same instance whenever a novel attack, also called zero day attack, is identified, a new rule can be incorporated into the database of rules with no extra effort. Remote attackers use vulnerabilities from Cross Site Scripting (XSS) that allows the attackers to infuse illogical web script through a crafted URL [9]. To handle the cross-site scripting vulnerability “SNORT-XSS” IDS system is used for XSS detection. The SNORT-XSS tool consists of five elements: In-Out Packet Decoder, Initial Processing, Recognition Engine, Alerting and Logging System, and lastly Output modules [10]. Table 2 presents the description of all five elements:

Table 2. Elements of SNORT-XSS tool.

Elements	Description
In-Out Packet Decoder	Extracts and categorizes the packets for further inspection.
Initial Processing	Integrates the procedure headers, recognize the abnormalities, rearrange the packets and examines the TCP flow.
Recognition Engine	Responsible for Pertain strategic rules to packets.
Alerting and Logging System	Responsible for Production of alerts & log messages.
Output modules	Exemption of alerts, actions for logging & creation of complete results.

#### 1.4. Utility of Soft Computing in IDS

Today's research on intrusion detection systems for network security uses advanced techniques mainly derived from soft computing. With the help of soft computing approach, solutions can be computed for existing complex problems, the output results may be rough or fuzzy in general. The foremost feature of soft computing is its adaptive nature, which means a change in circumstances does not influence the present method [11]. Some of the uniqueness of soft computing is listed below:

- It does not use any mathematical form for solving a problem.
- In different circumstances we get different solutions for the same problem with the same input values.
- It uses biologically stimulated techniques like genetics, particles swarming, evolution, human neuron system, etc.
- Adaptive in nature.

Three categories of soft computing methods are mainly used in general [12], namely:

- Fuzzy Logic.
- Artificial Neural Network.
- Genetic algorithm.

##### 1.4.1. Fuzzy Logic

Fuzzy Logic theory provides an arithmetical composition for defining and dealing with uncertainty, ambiguity, and predictable interpretation. The Fuzzy Logic concept is unlike conventional set theory. In conventional set theory, membership contribution is total or nil, but the fuzzy concept permits fractional membership sharing [13]. It can be explained as any element denoted as  $X$  that has a membership function denoted as  $\phi_f(X)$ , match to the level, for which  $X$  can fit into the set. Equation 1 presents the main membership function, which can be the set of universal elements:

$$\{ (X \in U) / \phi_f(X) = 1 \} \quad (1)$$

Equation 2 presents the predictable elements for the set:

$$\{ \text{For All } (X \in U) / \phi_f(X) > 0 \} \quad (2)$$

Operations (Union, Intersection, and Complement) from conventional set theory can also be implemented for a fuzzy set. Union of two distinct fuzzy sets  $f_1$  and  $f_2$  is a fuzzy set  $f_3$ , where  $f_3 = f_1 \cup f_2$  and Equation 3 presents the membership function:

$$\phi_{f_3}(X) = \text{MAX} (\phi_{f_1}(X), \phi_{f_2}(X)) = \phi_{f_1}(X) \vee \phi_{f_2}(X) \quad (3)$$

Intersection of two distinct fuzzy sets  $f_1$  and  $f_2$  is a fuzzy set  $f_3$ , where  $f_3 = f_1 \cap f_2$  and Equation 4 presents the membership function:

$$\phi_{f_3}(X) = \text{MIN} (\phi_{f_1}(X), \phi_{f_2}(X)) = \phi_{f_1}(X) \wedge \phi_{f_2}(X) \quad (4)$$

Similarly, the compliment of any fuzzy set  $f$ , can be represented as a fuzzy set  $\neg f$ , Equation 5 presents the membership function:

$$\phi_{(\neg f)}(X) = 1 - (\phi_f(X)) \quad (5)$$

Although combined rule-based efficient systems have been projected as the base for network intrusion detection and prevention systems, Fuzzy Logic rules is a superior method in the field of security. With the help of fuzzy

association rules one can propose a system that can be capable of extracting results from a known set of assessment data. These rules can be considered as high intensity to explain the samples of behaviors available in the data [14]. The objective of our research was to use a considerably remarkable way to mine the association rules. Lastly the defined rules are grouped using the concept of IF-THEN reasoning from Fuzzy Logic to ascertain an overall competent detection system.

#### 1.4.2. Artificial Neural Network (ANN)

Artificial Neural Network (ANN) uses the concept from human brain neural network's way of classification, that figures out in a dissimilar way apart from the traditional computer systems. As explained by Haykin, the human brain having the talent to classify its organization components, called as neurons, to carry out assured judgments like pattern identification and discrimination more rapidly than the most excellent digital computers [15]. To enhance the performance, artificial neural networks use massive interconnection of neurons. The significant Neural Network attains facts of the circumstances by a method called learning or training, which systematically adjust the interconnection effectiveness of the entire system in an organized way to attain an ideal design intention. To train the neurons of the neural network, a suitable training method is used. A combination of supervised, unsupervised and reinforcement learning techniques can be used to extract the features from obtainable models, with no knowledge of what are the outputs that are related with the chosen input samples, which means the learning technique recognizes and classifies the determined features with no responses from the surroundings. Unsupervised learning plays a great role in destructive neural networks to support data collection, feature mining, and identification of association. In artificial neural networks, one should take a note that the weighted vectors have the indistinguishable dimensionality as like the input sample. Each of the training iterations begins with the random choice of one of the input samples. Usually, the "Euclidean distance" between the weighted vector and the input sample is measured to decide which unit of neuron is to be activated at a particular moment in time. Generally, the neuron that has the smallest among the activation values is considered as the final neuron of the training iteration. At last, as per the requirements, the weight vectors of the final neuron and the selected neighborhood neurons are adjusted and fine-tuned. This process is carried out as iterations of continuing reduction between input sample and weight vector to get better results.

#### 1.4.3. Genetic Algorithm

The concept of genetic algorithm proposed by John Holland in 1965 is used to find solutions for problems that are based on evolutionary methods that use the ideology of natural selection [16]. They are generally used in optimization problems such as objective functions of maximization and minimization, which can be further classified into "ant colony" and "swarm particle" techniques. It follows the biological process of genetics and evolution. The genetic algorithm is capable of solving the problems that cannot be solved in real-time, which are termed as NP-Hard problems [17]. It uses the concept of randomized search and heuristic search method that provides a primary set of solutions and can generate a specific solution to the problem with a high accuracy rate.

The rest of the research article is structured as: 2<sup>nd</sup> Section focus on the literature review part, section-3 deals with the methodology adopted, section-4 emphasizes the performance evaluation of the proposed system and discussion of experimental results, and finally 5<sup>th</sup> section concludes the research work along with future scope.

## 2. LITERATURE REVIEW

A prompt, rule based misuse detection tool called SNORT, was created by Martin Roesch in the year 1998 and is now developed by Cisco Systems, specified in a particular open source language [18]. It is simple to insert the latest features or rules all the way through the period of compilation, which is one of the primary benefits of the tool. To collect the traffic packet data, SNORT makes use of shared object with an extension of .so files or simply

text files termed as rule files that are simple to comprehend and uphold. One can easily add new rules to the SNORT rules repository and can maintain it simply through notepads.

For organizing and to deal with uncertain and imprecise facts of real world, in 1965 Zadeh invented a technique called Fuzzy Logic, which today is one of the most proficient and widely used soft computing techniques [19, 20]. It is also a useful tool for decision making and for managing inaccurate and noisy data. An added advantage of using the fuzzy logic technique is that it can deal with uncertainty such as the time at which an alarm is to be raised is fuzzy, which is further helpful for classifications in abnormal conditions [21]. For detecting intrusions in a network a blended mode of Fuzzy Logic techniques were applied by many researchers in the field of Intrusion Detection and Prevention Systems [22, 23]. Mutual advantages of Artificial Neural Networks and Fuzzy Logic features were considered for the analysis of real time traffic. Similarly, based on Fuzzy Reasoning System, a three rule classifier soft computing NIDS was anticipated [24, 25]. A number of Network Intrusion Detection and Prevention Systems were projected by taking KDD Cup'99 and DARPA data set as input with the help of machine learning techniques like Artificial Neural Network (ANN), Genetic Algorithms (GA), Fuzzy Logic (FL) etc. [26, 27]. Even more sub classes of Artificial Neural Network techniques that comprise Grey Neural Networks, RBF, MLP, Recirculation Neural Networks and PCA are used for designing proficient intrusion detection systems that are mostly of misuse detection centric [28, 29]. To attain both misuse and anomaly detections, in recent times a number of researchers shared more than one technique like ANNs along with existing machine learning methods like GA or Fuzzy Logic.

### 3. PROPOSED MODEL - SNORT-XSS ALGORITHM

In the beginning, rule sets were produced for SNORT-XSS tool, by considering KDD Cup'99 and DARPA dataset, and then these sets were categorized into a few Fuzzy sets with the help of Fuzzification method and Fuzzy IF-THEN rules and finally these rules were used for training, validating and testing the proposed system with the help of Artificial Neural Networks for classification and recognition of real time intrusions. Figure 1 illustrates the simple structure of SNORT-XSS rules and rule application order for SNORT-XSS can be:

*pass → drop → sdrop → reject → alert → log*

```

"alert icmp any any -> any any (msg:"Testing ICMP"; sid:1000001;)"
"alert tcp any any -> any any (msg:"Testing TCP"; sid:1000002;)"
"alert udp any any <-> any any (msg:"Testing UDP"; sid:1000003;)"
"alert tcp any any -> [a.b.0.0/06, c.d.e.0/14] 90 (msg:"ATTACKS attempt"; nocase; sid: 1384; flow: to_server, established; content:" "; [...])"
    
```

Figure 1. Rule structure of SNORT-XSS.

The 1<sup>st</sup> field from the SNORT-XSS rule structure indicates the action in terms of response, which can be Block, Alert, Terminate, Activate or Log. The second field in the rule represents the protocol that is currently used, which can be UDP, TCP and/or ICMP. Subsequently the 3<sup>rd</sup> and 4<sup>th</sup> fields of the rule set specifies the source IP and corresponding port number that is associated with it. The field “any any” in the rule set specifies that the incoming packets are from any source IP and from any associated port number. The 5<sup>th</sup> field describes whether the transmission is unidirectional represented as “→” or bi-directional represented as “<>” followed by destination IP and port number, and finally a message for the record purpose. The SNORT-XSS tool can be initiated running in packet dump mode as mentioned below:

```

C:\Snort\bin>snort -v
Running in packet dump mode
  --- Initializing Snort ---
Initializing Output Plugins!
    
```

pcap DAQ configured to passive.

The DAQ version does not support reload.

Acquiring network traffic from "\Device\NPF\_{E33433AF-4E43-47AC-9D91-C34EB3E652C6}".

Decoding Ethernet

--== Initialization Complete ==--

-\*> Snort! <\*-

Version 2.9.19-WIN64 GRE (Build 85)

By Martin Roesch & The Snort Team: <http://www.snort.org/contact#team>

Copyright (C) 2014-2021 Cisco and/or its affiliates. All rights reserved.

Copyright (C) 1998-2013 Sourcefire, Inc., et al.

Using PCRE version: 8.10 2010-06-25

Using ZLIB version: 1.2.11

Commencing packet processing (pid=636)

In the rule structure, one of the interfaces is to be selected from the currently available interfaces. The tool can then be run in test mode using the command as “snort -i 1 -c c:\Snort\etc\snort.conf -T”. The rule chains will be initialized as below:

+++++

Initializing rule chains...

10279 Snort rules read

9835 detection rules

153 decoder rules

291 preprocessor rules

10279 Option Chains linked into 300 Chain Headers

+++++

-----[Rule Port Counts]-----

	tcp	udp	icmp	ip
src 3654 23	0	0		
dst 5841 74	0	0		
any 684	3	4	0	
nc 453 1	1	0		
s+d 4 2	0	0		

The port based pattern matching for SNORT-XSS tool is given below:

[ Port Based Pattern Matching Memory]

+- [ Aho-Corasick Summary ] -----

| Storage Format : Full-Q

| Finite Automaton: DFA

| Alphabet Size : 256 Chars

| Size of State : Variable (1,2,4 bytes)

| Instances : 205

| 1 byte states : 192

| 2 byte states : 12

| 4 byte states : 1

| Characters : 208840

| States : 165829

```

| Transitions : 29441909
| State Density : 69.4%
| Patterns : 9929
| Match States : 10240
| Memory (MB) : 118.76
| Patterns : 1.16
| Match Lists : 2.61
| DFA
| 1 byte states : 1.06
| 2 byte states : 47.46
| 4 byte states : 66.12
+-----+
[ Number of patterns truncated to 20 bytes: 561 ]

```

Finally, the tool should be running in IDS mode for validation. For the entire research SNORT-XSS Rules Engine: SF\_SNORT\_DETECTION\_ENGINE Version 3.2 <Build 1> was used.

Using Fuzzy IF-THEN Reasoning, the defined snort rules are then classified into five distinct Fuzzy sets, like: “Very Low (VL)”, “Low (L)”, “Medium (M)”, “High (H)” and “Very High (VH)”, so that the accuracy rate of true negative and false positive is high with reduced false alarm rate. A simple Fuzzy reasoning IF-THEN-ELSE policy was applied for this purpose and which can be represented as below:

*IF Feature\_1 is LOW and Feature\_2 is HIGH, THEN output is Regular ELSE output is Intrusive.*

The aptitude of Fuzzy rationale in Intrusion Detection System has the additional outline as below:

```

IF      Conditions
THEN    Consequences

```

Where conditions are the fuzzy, variables and consequences are the fuzzy sets.

The Fuzzy logic concept presents the simplest method of reaching a particular conclusion based on unclear, noisy, ambiguous, missing and imprecise input data. The steps followed for reaching a specific conclusion is demonstrated below.

- All input featured variables are fuzzified or categorized into membership functions.
- Fuzzy policies are generated. The policies are in terms of IF-THEN statements.
- At a particular instance, few of the fuzzy policies will be triggered.
- The policies that are activated will be combined in the rules repository to estimate the fuzzy output distribution.
- Finally, the fuzzy output distribution will be “defuzzified” to get a combined output value.

The concept necessary to fuzzy logic systems is that the truth values in fuzzy reasoning and or membership values in fuzzy rational sets are represented by a value that range between 0 and 1,

$$0.0 \leq \text{Membership\_Values} \leq 1.0$$

With 0.0 indicating absolute falseness, and 1.0 indicating absolute truth. A sample rule, its relation and membership function associated is described below:

Sample Rule: *IF X is A THEN Y is B ELSE Y is C*

Sample Relation:  $R = (A \times B) \cup (\bar{A} \times C)$

Sample Membership Function:

$$\phi(X, Y) = \text{MAX} [ \text{MIN} \{ \phi A(X), \phi B(Y) \}, \text{MIN} \{ 1 - \phi A(X), \phi C(Y) \} ]$$

For example, initially the NIDS examines the packets and determines the packets in whole with the same intention value to be 40, then this incidence is to be considered as LOW Fuzzy set for a degree value of 0.4.

However, for a degree value of 0.6 the same is to be considered as VERY HIGH Fuzzy set. Figure 2 illustrates this experience of fuzzy set organization.

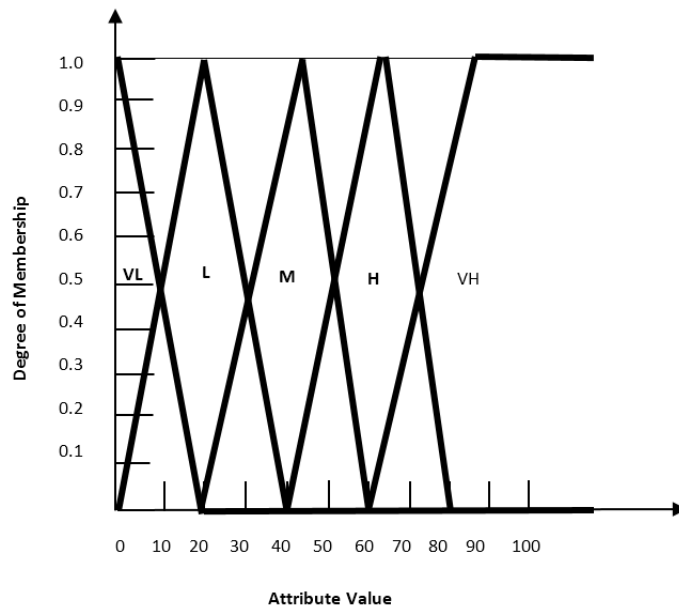


Figure 2. Fuzzy break used for snort rule classification.

### 3.1 Fuzzy Rule Coding

We have considered linguistic set of five variables and one don't-care condition as the predecessor fuzzy sets. Only the antecedent part of each of the IF-THEN rule is coded as an individual. These linguistic fuzzy sets are:

- 1: Very Low (VL).
- 2: Low (L).
- 3: Medium (M).
- 4: High (H).
- 5: Very High (VH).
- δ: Don't Care (DC).

For example a rule is coded as “2-δ-5-δ” and can be treated as IF X1 is Low and X2 is Don't Care and X3 is Very High and X4 is Don't Care THEN followed by consequences.

Once we are ready with the Fuzzy policy sets, we can Train, Validate and or Test the system. For this purpose, “Back Propagation of Errors with Feed Forward Artificial Neural Networks” was used. Figure 3 illustrates the design of feed forward ANNs with back propagation of errors.

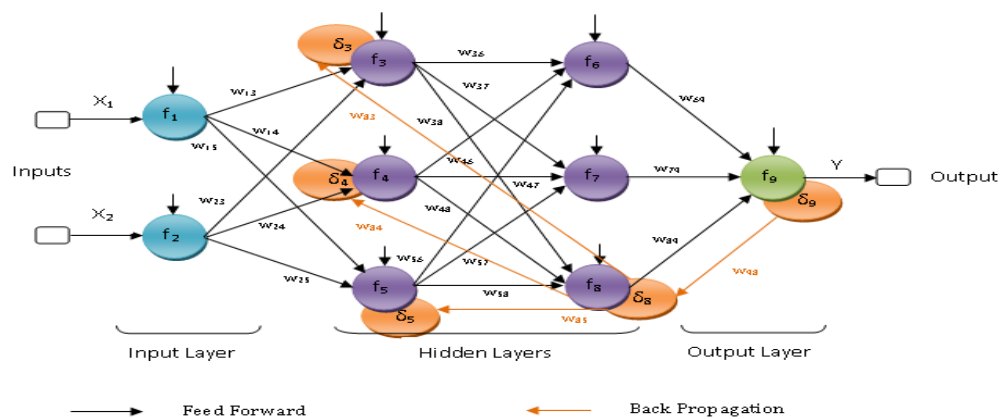


Figure 3. Design of feed forward ANNs with back propagation of errors.



### 3.2 Training Algorithm

The entire training process of Feed Forward ANNs with Back Propagation of Errors consists of four phases, such as:

Phase-1: Initialization of neuron Weights.

Phase-2: Feed Forward of outputs to next set of neurons.

Phase-3: Back Propagation of final errors.

Phase-4: Updation of the neuron weights and fine tuning the biases.

Based upon the proposed model, the final output  $Y$  can be calculated by Sum of Products (SOP) described as below:

$$Y = f_9 (W_{69}Y_6 + W_{79}Y_7 + W_{89}Y_8) \quad (6)$$

The back propagation of error  $\delta_5$  to adjust and fine tune the neuron  $f_5$  can be calculated as below:

$$\delta_5 = W_{98}\delta_9 + W_{85}\delta_8 \quad (7)$$

Figure 4 illustrates the entire proposed system. It starts with SNORT-XSS rules generation from standard KDD CUP'99 and DARPA data set. Then those generated SNORT rules are fuzzified into different fuzzy sets using Fuzzy IF-THEN reasoning. After preprocessing the fuzzified rules, the system is trained, validated and lastly tested with the help of Feed Forward ANNs with Back Propagation of Errors by considering real time intrusive data for evaluating the system.

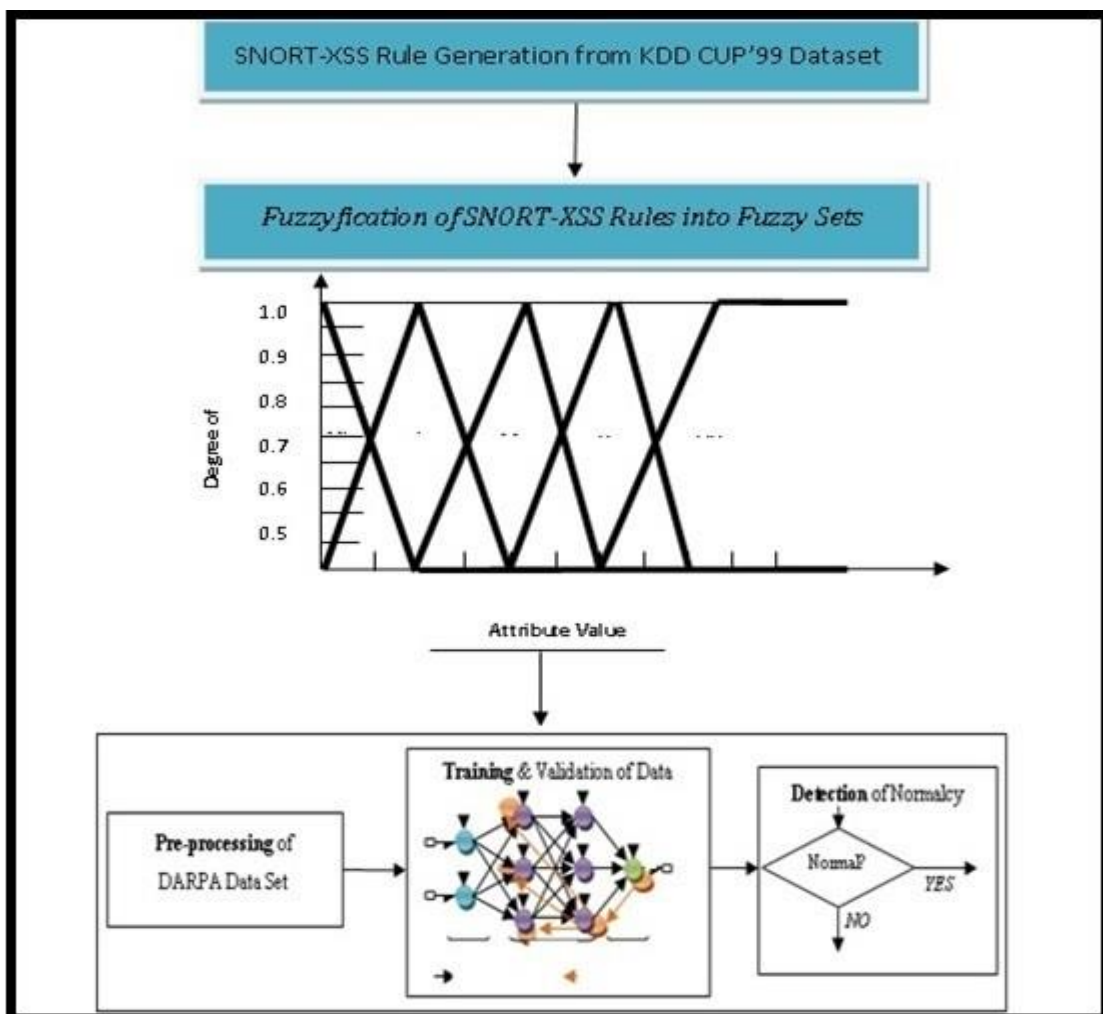


Figure 4. Proposed overall SNORT-XSS based NIDS system.

#### 4. EXPERIMENTAL RESULTS AND DISCUSSION

Almost all existing NIDS make use of exclusive data as input, for which consequences are not repeatable and are not testable. To overcome this drawback, we used standard DARPA and KDD CUP'99 data set in our research experiment. Apart from this, one more concern for assessment of IDS is then on-availability of the real time intrusive data. In our research, almost all the available instances of DARPA and KDD Cup'99 dataset was taken into consideration for training and testing the proposed system. Table 3 presents the training and testing wise rule set composition that includes the types of attacks:

Table 3. KDD Cup'99 Training and Testing set composition.

Instance Type	Training instances	Testing instances
Normal	98168	61683
DoS	382567	228683
Probe	4218	4267
R2L	1037	8714
U2R	56	72
Total	486046	303419

The proposed system, along with SNORT-XSS, is tested and validated upon real time network traffic and then attacks that are identified are examined in a regular basis. Figure 5 illustrates the results based on % of damage due to the attack (Figure 5A), false alarm rate (Figure 5B), precision in attack detection (Figure 5C) and transmission of mean squared error (Figure 5D)for the system.

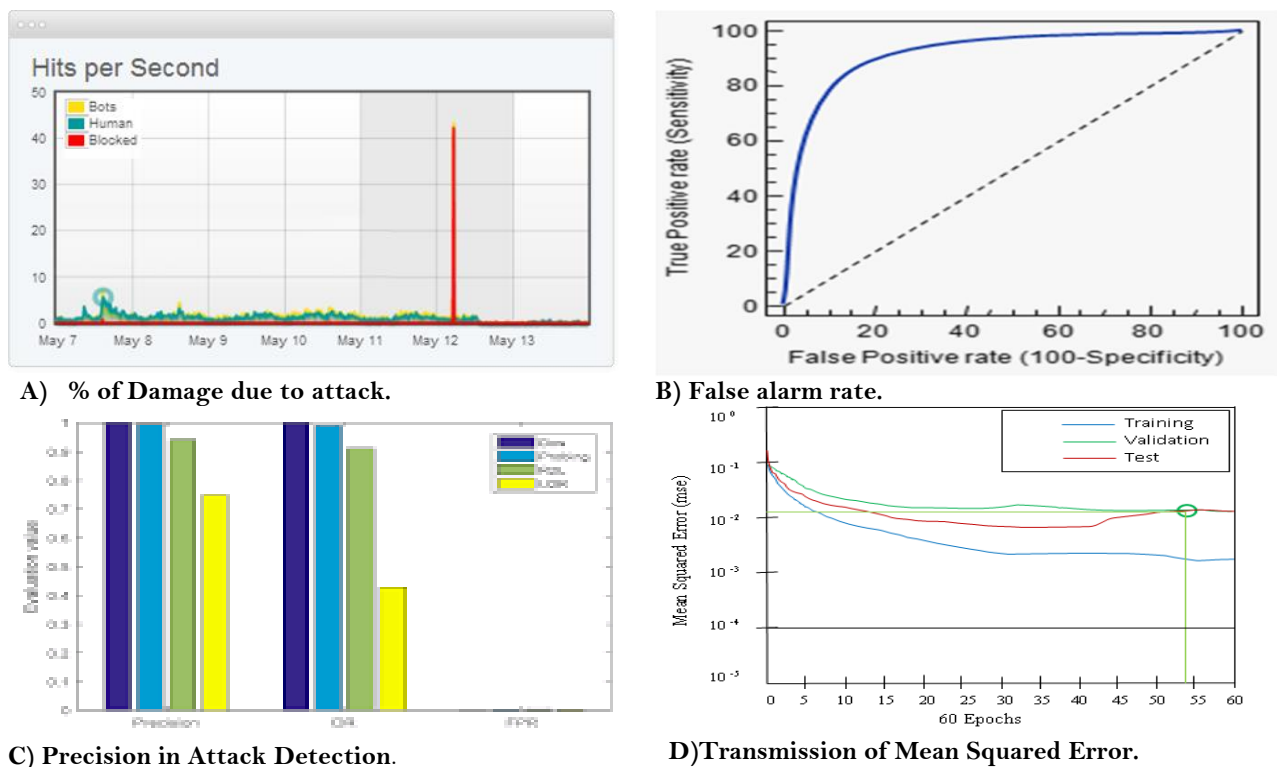


Figure 5. Experimental result analysis of proposed Snort-XSS based IDS.

The system was tested with real time network traffic for a period of one week. The percentage of damage to the system due to attacks by human, bots and blocked intrusions were analyzed and are plotted in the above graph. We can observe that the damage is below 3%. Here one thing to be noted is that during training the system goes offline and while the system is offline the percentage of damage is bit high, which is the drawback of the proposed system and can be focused on the near future. The assessment parameters of all types of attacks are as follows:

the Precision values of the proposed model were 98.93% and 98.89% respectively, and detection rate of Probe and DoS attacks were greater than 98%. The false positive rate is almost negligible as shown in the graph. From the mean squared error transmission graph it can be identified that the finest classification was acquired at epoch numbers 50 to 55 with a mean squared error of 0.004, where test, training and validation share a greatest common minimum. Outstanding results were achieved with the help of Fuzzy IF-THEN reasoning for coding and categorizing rule sets, ANNs with error back propagation for training, testing and validating the proposed system by considering KDD Cup'99 dataset as input derived from DARPA.

## 5. CONCLUSION & FUTURE SCOPE

Network Intrusion Detection Systems should have some sort of intelligence to identify attacks and should respond immediately. For this purpose, the system should be well trained, tested and validated. Because of this, standard KDD Cup'99 dataset derived from DARPA was taken into consideration for the proposed system. The open source, light weight and easy to use SNORT-XSS tool provided by CISCO Systems was used to attain an effective intrusion detection and prevention system. The policies that were produced from the SNORT-XSS tool were classified into a number of Fuzzy sets with the help of Fuzzy reasoning and IF-THEN policies to minimize the false alarm rate. Lastly, the fuzzified rules were used for training and testing the anticipated system using Feed Forward ANNs with Back Propagation of Errors and the achieved outcomes were as promising. From the experimental results and discussion section we can conclude that outstanding outcomes of the proposed IDS system were achieved.

In future, one can come up with even influential statistical techniques in combination with the SNORT open-source tool for better accuracy in intrusion detection, by virtue of which some of the existing issues can be addressed, such as:

- Detection system slows down due to voluminous rule sets and patches that are available day by day. Research can be carried out in this direction to optimize the policy sets into the minimum possible.
- Training period takes the detection system offline. During this time period percentage of damage to the system may be high.
- At the point of attack initiation itself, it should be detected and preventive measures should be taken to reduce the harm to the system and the performance can be enhanced even more.

**Funding:** This study received no specific financial support.

**Competing Interests:** The authors declare that they have no competing interests.

**Authors' Contributions:** All authors contributed equally to the conception and design of the study.

## REFERENCES

- [1] S. Rathore, A. Saxena, and M. Manoria, "Intrusion detection system on KDDCup99 Dataset: A survey," *International Journal of Computer Science and Information Technologies*, vol. 6, pp. 3345-3348, 2015.
- [2] R. Shanmugavadivu and N. Nagarajan, "Network intrusion detection system using fuzzy logic," *International Journal of Computer Science and Engineering*, vol. 6, pp. 19-23, 2011.
- [3] S. Mishra, S. K. Pradhan, and S. K. Rath, "Network intrusion detection system using fuzzy if-then rules and fuzzy reasoning: A soft computing technique " *International Journal of Computer Engineering and Applications*, vol. 12, pp. 238-245, 2018.
- [4] S. Mishra, S. K. Pradhan, and M. Pradhan, "Intrusion detection and prevention system for zero-day attacks: A two dimensional approach," *International Journal of Applied Engineering Research*, vol. 10, pp. 10767-10782, 2015.
- [5] G. Nadiammai and M. Hemalatha, "Snort based network traffic anomaly detector to improve the performance of intrusion detection system," *International Journal of Advanced Research in Computer Science*, vol. 3, pp. 9-13, 2012.

- [6] M. Tavallae, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," presented at the Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.
- [7] S. Mishra, S. K. Pradhan, and S. K. Rath, "Performance analysis of network intrusion detection system using back propagation for feed forward neural network in MATLAB/SIMULINK," *International Journal of Computational Engineering Research (IJCER)*, vol. 8, pp. 58-65, 2018.
- [8] R. Martin, "SNORT users manual-2.9.16. Retrieved from: [https://snort-org-site.s3.amazonaws.com/production/document\\_files/files/000/000/249/original/snort\\_manual.pdf](https://snort-org-site.s3.amazonaws.com/production/document_files/files/000/000/249/original/snort_manual.pdf). [Accessed January, 2022]," 2022.
- [9] P. S. Bhattacharjee and S. A. Begum, "Fuzzy approach for intrusion detection system: A Survey," *International Journal of Advanced Research in Computer Science*, vol. 4, pp. 101-107, 2013.
- [10] N. N. P. Mkuzangwe and F. V. Nelwamondo, *A fuzzy logic based network intrusion detection system for predicting the TCP SYN flooding attack*. Kanazawa, Japan: Lecture Notes in Artificial Intelligence LNCS/LNAI, 2017.
- [11] D. N. P. Suthishni and G. P. Ramesh, "Intrusion detection analysis by implementing fuzzy logic," *International Journal of Applied Engineering Research*, vol. 11, pp. 3216-3220, 2016.
- [12] A. H. Selman, R. Koker, and S. Selman, "Intrusion detection using neural network committee machine," *International Conference on Information, Communication and Automation Technologies*, vol. 9, pp. 238-247, 2013.
- [13] S. Mishra, S. K. Pradhan, and S. K. Rath, *Network intrusion detection system using soft computing technique - fuzzy logic versus neural network: A comparative study*, *Cognitive informatics and soft computing, Advances in Intelligent Systems and Computing 1040*. Singapore: Springer Nature Singapore Pte. Ltd, 2020.
- [14] T. Pandit and A. Dudy, "A feed forward artificial neural network based system to minimize Dos attack in wireless network," *International Journal of Advances in Engineering & Technology*, vol. 7, p. 938, 2014.
- [15] R. P. Lippmann and J. Haines, "Analysis and results of the 1999 DARPA off-line intrusion detection evaluation," in *Proceedings of Third International Workshop on Recent Advances in Intrusion, RAID 2000, LNCS 1907, 2000*, pp. 162-182.
- [16] A. S. Sodiya, O. A. Ojesanmi, and O. C. Akinola, "Neural network based intrusion detection systems," *International Journal of Computer Applications*, vol. 106, pp. 19-24, 2014.
- [17] P. J. Patel, J. S. Shah, and J. Patel, "Performance analysis of neural networks for intrusion detection system," *International Journal of Computer Technology and Applications*, vol. 8, pp. 88-93, 2017.
- [18] R. Martin, "Snort-lightweight intrusion detection for networks," presented at the 13th Systems Administration Conference, 1999.
- [19] S. Mishra, S. K. Pradhan, and S. K. Rath, "Detection of zero-day attacks in network IDS through High performance soft computing," presented at the International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021.
- [20] R. Yao, N. Wang, Z. Liu, P. Chen, and X. Sheng, "Intrusion detection system in the advanced metering infrastructure: A cross-layer feature-fusion CNN-LSTM-based approach," *Sensors*, vol. 21, p. 626, 2021. Available at: <https://doi.org/10.3390/s21020626>.
- [21] B. Sujatha and V. Kavitha, "Survey on intrusion detection approaches," *International Journal of Advanced Research in Computer Science*, vol. 3, pp. 363-371, 2012.
- [22] A. Ranjan, D. R. S. Hegadi, and P. Kumara, "Emerging trends in data mining for intrusion detection," *International Journal of Advanced Research in Computer Science*, vol. 3, pp. 279-281, 2012.
- [23] G. D. Kurundkar, N. A. Naik, and S. D. Khamithar, "Network intrusion detection using snort," *International Journal of Engineering Research and Application (IJERA)*, vol. 2, pp. 1288-1296, 2012.
- [24] J. Gomez, C. Gil, N. Padilla, R. Barios, and C. Jimenez, "Design of a snort-based hybrid intrusion detection system," presented at the In International Work-Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg, 2009.
- [25] S. Devaraju and S. Ramakrishnan, "Performance comparison for intrusion detection system using neural network with KDD Dataset," *ICTACT Journal on Soft Computing*, vol. 4, pp. 743-752, 2014.

- [26] M. Gyanchandani, R. N. Yadav, and J. L. Rana, "Intrusion detection using C4.5: Performance enhancement by classifier combination," *ACEEE Int. J. on Signal & Image Processing*, vol. 1, pp. 46-49, 2010.
- [27] M. Mostaque and M. Hassan, "Network intrusion detection system using genetic algorithm and fuzzy logic," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, pp. 1435-1445, 2013.
- [28] S. Mahmudova, "An examination of the methods of increasing software efficiency based on soft computing technology," *Review of Computer Engineering Research*, vol. 6, pp. 45-56, 2019. Available at: <https://doi.org/10.18488/journal.76.2019.61.45.56>.
- [29] R. M. Alguliyev and M. S. Hajirahimova, "Classification ensemble based anomaly detection in network traffic," *Review of Computer Engineering Research*, vol. 6, pp. 12-23, 2019. Available at: <https://doi.org/10.18488/journal.76.2019.61.12.23>.

*Views and opinions expressed in this article are the views and opinions of the author(s), Review of Computer Engineering Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.*