Novel Anomaly Intrusion Detection Using Neuro-Fuzzy Inference System

K.S. Anil Kumar and Dr. V. NandaMohan

Sree Ayyappa College, Alappuzha, Kerala, India

Summary
Conventional approaches to intrusion detection system pose a myriad of problems that exhibit serious impediments to the degree of configurability, extensibility, and effectiveness of the systems. The proposed methodology is a combination of three techniques comprising two machine-learning paradigms. K-Means Clustering, Fuzzy Logics and Neural Network techniques deployed to configure an effective intrusion detection system. Out of the several problems in the traditional techniques of Intrusion Detection Systems, the presence of high rate of false alerts causes unnecessary interference of human analyst. The human analysts in turn perform an intensive analysis repeatedly to distinguish the nature of such alerts and initiate sufficient actions. The approach proposed reveals the advantage of converging K-Means – Fuzzy – Neuro techniques to eliminate the preventable interference of human analyst in such occasions. The technique was tested using multitude of background knowledge sets in DARPA network traffic datasets. The experimental results render remarkable improvement in reducing the false alarms in addition to increased ability to capture intrusion packets that are no similar to the ones in the training datasets.

Key words:
Intrusion Detection System (IDS), Anomaly Detection, Neuro-Fuzzy, DARPA data set.

1. Introduction

The proliferation of computer networks in the contemporary period, in particular the contribution of e-commerce to the world economy, has necessitated the security of computer networks an international priority. Intrusion detection has become an inevitable area of research, since it is technically infeasible to build a system that can offer total resistance to attacks. Intrusion as generally described is an act of trespassing or infringing the integrity, confidentiality or preventing the availability of a resource [1]. Intrusions Detection Systems detects unauthorized or malicious attacks over a computer system which occurs primarily through internet. These attacks can compromise the security and trust of a system. These attacks can take several forms like network attack against vulnerable services, data driven attacks on applications, host based attacks such as privilege escalation, unauthorized logins and access to sensitive files. Categorized broadly based on their patterns of detection, IDSs can be classified as misuse detectors or anomaly detectors. Misuse detectors rely on comprehending the patterns of known attacks [2, 3], while anomaly detection exploits user profiles as the basis of detection, and brands the characteristics of the deviant from the normal ones as intrusion [2, 3, 4, 5]. Generally, IDSs comprise several components that include Sensors, Console and Engine. Sensors generate the security events; Consoles monitor events and alerts and control the sensors. Engines maintain record of events logged by the sensors in a database and use it as protocol to generate alerts from security events received.

The concept we proposed works in a similar manner and exploits various algorithms apt for fulfilling the requirements at every stage. We have used K-Means to cluster normal and intrusion packets, Fuzzy logic for rule generation from the perceived traits of normal and abnormal clusters, and Neural Networks for detecting abnormal packets similar to the ones given during training sessions and for an artificial intelligence that detects anomalies not presented during training [6].

2. Preliminaries

This section gives a brief introduction of the techniques used in the proposed work.

2.1 K- Means Clustering Algorithm

K-means clustering is a technique that classifies objects in to K number of groups based on their attributes or features. Obviously K is a positive number. The cluster centriod is calculated first. Then the grouping is done by minimizing the sum of squares of distances between the data and the corresponding cluster centroid. The object of K means clustering is to classify the data by analyzing the traits and then organizing them in accordance to their attributes. The reason why K-Means clustering has been chosen for the algorithm is that the packets we analyze needs to be categorized in to just two clusters, normal and intrusion. And hence the value of K can be simply defined as '2' [7].

Manuscript received August 5, 2008.
Manuscript revised August 20, 2008.

2.2 Fuzzy Logic

Out of the theory of fuzzy sets developed by Lotif Zadeh in 1965, the term “fuzzy logic” originated [8]. The term exhibits greater degree of compatibility when dealing with the data or objects that are abstract in nature. To illustrate, when we say that someone is tall or short, we don’t specify the exact height of the person. It is intuitively understood that the person is somewhat taller or shorter with respect the perceived average height. Hence for a fuzzy subset A of set X is distinguished by assigning to each element x of X the degree of association of x in A. For example X is a group of people, A the fuzzy set of tall or short people in X. If we consider X as a set of propositions then its elements may mete out their extent of truth, which could be “absolutely true”, ”absolutely false” or some intermediate degree of truth i.e., the proposition may be truer than another proposition. These forms of expressions are evident while describing propositions like “this guy is tall” (handsome, rich, and etc).

Here in our algorithm we use the same method for segregating normal from the abnormal. The degree to which the attributes of data packets impact the decision of abnormality or normalcy are represented in terms of weight ages assigned [9].

Now a question that arises is how the propositions can be connected by connectives (conjunction, disjunction, negation etc) within the analogy of diverse definitions of operations on fuzzy sets like (intersection, union, complement…) and whether the truth degree of the composed proposition is influenced by the truth degree of the constituent components, to be exact whether the connectives have their equivalent truth functions (one like the logic truth table). If this concept is accepted, the truth functional approach; it makes fuzzy logic to be completely different from the probability theory, as probability theory is not truth functional. [11] (The constituent propositions do not determine the probability of conjunction of propositions). Since it wouldn’t be enough to rely on probability theory alone, the algorithm we devised overwhelms this aspect by scrutinizing every possible combination of weight ages through an application of a random function. This function generates training sets so meticulously that it leaves no possible combinations go unchecked. Despite fuzzy logics are inherently based on probabilistic theory the approach can help emulate a truth functional approach.

2.3 Artificial Neural Network (Ann)

The architecture of neural network comprises a system of programs and data structures that resemble the function and structure of human brain. Neural networks can learn, differentiate, and extract the underlying correlations and sort out the patterns of the data fed. Generally executed in conventional computers, neural networks typically are software simulations. They focus on structuring the connections between the processing elements termed as neurons by the terminology rather than manipulating zeroes and ones for computation as performed in the conventional digital modal. These structures and weights determine the output. Hence we factored in this aspect of the neural network mechanism; and we assigned weights to the fuzzy rules generated. These weights denote the degree to which the presence of a particular attribute has influenced the conclusion of abnormality of the intrusion packet. Figure 1 is the logical structure of Neural Network.

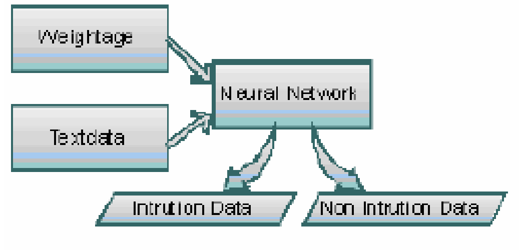


Fig. 1 Logical Organization of the Deployed Neural Network

Typically neurons in the network have one or more inputs and the weights associated with these inputs determine the importance of values that are fed to the corresponding neuron. Neural networks designate these neurons to manipulate the inputs mathematically and generate the output. Here is where we perform the analysis of the fuzzy rules under scrutiny, with reference to their calibrated weights. This analysis is performed in every neuron in the network until the final results are obtained. Neural networks are known to exert optimum performance; proportionate to the amount of training it receives. To extract a precise or at least a near precise recognition, the solution we proposed adjusts the composition of weights randomly covering all possible values to train the system. The system can now be expected to initiate its own activity in response to external stimuli not exactly matching the ones presented during the training sessions [10]. Figure 2 depicts the organization of neuron Neural networks can deploy a myriad of approaches for making determinations that include genetic algorithm, fuzzy logic, gradient based training, and Bayesian methods. After a comprehensive analysis we opt to make use of fuzzy logic in the algorithm for rule generation. Based on the complexity of the requirement, layers (called occasionally as knowledge layers) in neural networks can be organized in varying quantities and the system can be graded accordingly. The solution uses the built in facility of Matlab to feed forward the learned relationship to higher knowledge layers.

Started with a theoretical estimation of improved performance for intrusion detection, through congregating the techniques of neural network, fuzzy and K-means, the architecture proves to be fast and yields close approximation of the expected results in experimental verification. The technique excels in recognizing patterns of abnormality, learning from experience, and sorting out the normal from the abnormal.

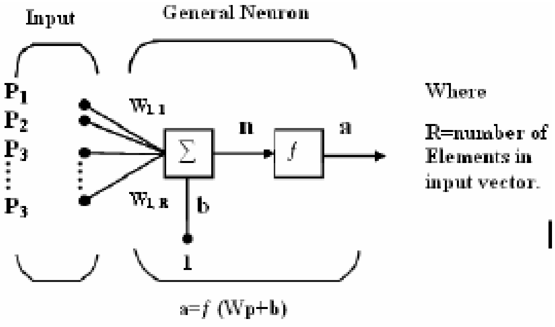


Fig. 2 General Representation of a Neuron

3. Proposed Methodology

The proposed solution atempts to exploit the potential of K-means Clustering, Fuzzy Logic and Nueral Network techniques for Intrusion Detection. Effeiciency of the proposed solution is analysed by sampling DARPA datasets. The fields in the datasets are scrutinized and categorized as intrusion and normal packets. This classification is done by applying K-means clustering technique. The inherent defieciency of the K-Means clustering is to initialise the value of K, a number that determines the count of clusters to form. As the case we are dealing with has just two segregations ie., normal and abnormal, the value of K can be initialized to two, simply. It is generally considered that the volume of intrusion packets are less than that of normal packets. Using this aspect of intrusion as a criteria, we perform the partitions by bifurcating the normal packets from the abnormal ones.

Once after this partitions are performed every field in the normal and deviant cluster is probed to understand its characteristics. Analysing the intrinsic nature of these intrusion packets can give a clear picture of the factors that signify the abnormalcy. This knowledge can futher help to distinguish the regular from the irregular ones. A similar process is carried out in parallel to interpret the patterns of normalcy from the regular packets.

To aid this process the algorithm proposed, formulates a fuzzy rule using the knowledge gained through analysis, that it can help discrimate the regular and intrusion packet. The fuzzy rule places all the normal and abnormal packets in a separate set or a vector. And hence when a packet is received the fuzzy logic itself can classify the regular packet from the deviant ones.

The impact of every rule thus formed can be calibrated and weights assigned. The weightages thus arrived become the training pattern for Neural Network technique.

The algorithm initiates the work by performing a thorough analysis of the intrusion and normal data packets. The list of the following fields Type, Count, Land, and Srv_rate from intrusion or abnormal packets are collected with their corresponding data values and interpreted from the table that accumulated it.

SQL queries are deployed to shortlist the distinct values contained in each of those fields interpreted in the previous step. This process results in creation of a table of contents, hosting an assemblage of uniquely selected of data values. And these values symbolize the characteristics of abnormality in the intrusion packets and normalcy from the regular packets.

The subsequent process begins by calculating the count of such records amassed, and then it prepares a sorting of maximum combination of the short-listed values. To achieve this amalgamation, an inbuilt function for random number generation is applied. The function by its very nature prepares a permutation and combination of these values so as no possible mix of the combinations is missed.

The algorithm then computes the percentage of presence of each of those combinations to draw an understanding of how far a particular combination is present in the intrusion dataset. This step can help yield the knowledge of the extent to which each specific combination of these characteristics influence the degree of intrusion.

These percentages thus derived are christened as weight ages and stored separately in a database. We have chosen to exploit neural network technique and deployed back propagation algorithm. Since back propagation algorithm is most suitable, as it can learn from weighted input data. To nurture an intelligence that can help detect intrusion packets from the normal packets, the technique is trained to identify the presence of characteristics with respect to the weights they carry that signifies intrusion.

Furthermore, to reinforce the process of segregating the normal from the intrusion and to reduce the false alarm rate the algorithm performs the same sequence of steps but with an object to identify the patterns of normality or the characteristics of regular packets. This process is done in parallel acquiring the training datasets from the normal cluster initially obtained through K-Means Clustering.

After the training meant for learning the traits of normal and intrusion packets are completed, the solution can be deployed for capturing intrusion in a real time environment. The reason the algorithm is designed to perform dual learning of the characteristics of abnormal versus normal is to overwhelm the possibility of indecisiveness and the subsequent wrong predictions due to nuances that exist in their patterns. Invalidating the need for two-way analysis of the archetypes of normal as well as deviant can lead to ineffectual prevention of intrusion packets at the same time higher rates of false alarms. Figure 3 portrays the steps of the algorithm.

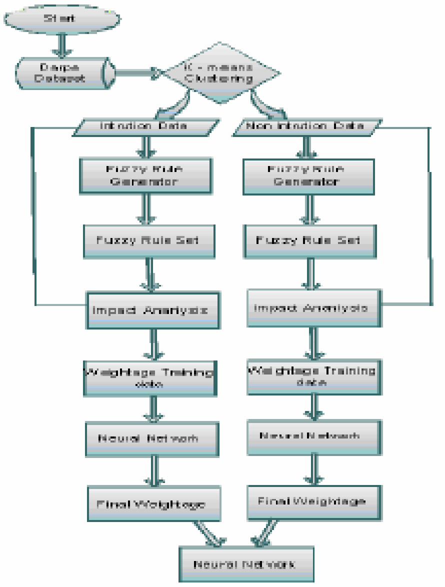


Fig. 3 An illustration of the sequence steps in the algorithm

Needless to mention, techniques of neural networks are capable of detecting intrusion packets not only those that exactly matches the characteristics of the training sets but also the ones that resemble such traits. We have evidential proof, collected through experimental research that our algorithm excels in identifying synthetically generated intrusion packets, which are dissimilar to those of the training datasets. The algorithm also offers significant reduction in false alarm rates and eliminates the need for interference of a human analyst.

Here is the representation of the pseudo code for the above-mentioned operations

Initialize Increment to 1
Initialize Wight of Find Record to 0
Initialize Qcnt to 1

NI	→	Number of iteration
FL	→	Field Length
$QStr$	→	Query String
IL	→	Index Length
$TotFR$	→	Total Record Find
$TotR$	→	Total Record in Database
Wht	→	Wight of Find Record
$Qcnt$	→	Query Find Count
Nfl	→	Vector [FL] length

```
WHILE Increment < NI
    FOR each value FL
        $Index[FL] = rand() \bmod Nfl$
    ENDFOR
    FOR each value IL
        $QStr = sql$ select statement where each
        $Field[IL] = Index[IL]$
    ENDFOR
        $TotFR = ExecuteQuery(Qstr)$
    IF TotFR is non zero THEN
        $Wht[Qcnt] = TotR / TotFR$
        Add 1 to Qcnt
    ENDIF
        Add 1 to Increment
ENDWHILE
Save Wht
Save Qstr
```

Over the years, combination of techniques of neural network and fuzzy logic are pervasively applied for solving problems in information processing. These methods have their own advantages in dealing problems; especially vague or imprecise data can be processed using fuzzy logic while neural network techniques fare better in dealing with near precise or precise inputs. On the other hand these two techniques have serious setbacks. In particular, fuzzy logic is not good enough for handling accurate information; it works by generalizing the inputs dealt. Neural networks demand exactitude of the inputs for better analysis and prediction.

Combining these algorithms can yield the synergistic power of both and help overwhelm the inherent deficiencies of these techniques. Our approach over intrusion detection aims to exploit the convergence of these methodologies. Initially we performed K-means clustering to segregate the sampled DARPA dataset in to normal and abnormal. The datasets classified as abnormal are then analyzed thoroughly to interpret the attributes that symbolize the abnormality. Based on the interpretation thus arrived we generate the fuzzy rule. The reason why fuzzy logic is applied is because the interpretations made cannot be considered precise. Fuzzy logic comes handy when dealing with immense volume of imprecise or probabilistic data. In our scenario there exist just two categories of datasets, normal and abnormal. Constructing just two fuzzy sets or fuzzy vectors to hold the normal and abnormal datasets respectively is simple and effective.

Now we have an accumulation of characteristics of all possible abnormal packets congregated from the DARPA dataset. The fuzzy rules engendered out of it would be enough for a rough discrimination of normal from the abnormal packets. But to refine the effectiveness of the solution proposed, to yield a near precise if not precise isolation of abnormal packets in network, separate weight ages are assigned to every rule.

The weight ages are calculated based on an appraisal of the presence of certain attributes in the abnormal and normal categories of the datasets and the volume of its presence. In a nutshell these weight ages signify the impact of every rule and conversely it represents the degree to which the rule can help identification and isolation of abnormal packets in network.

Neural Networks has established its reputation due to its efficiency in dealing with tasks that require considerable levels of precision in prediction. The algorithm mimics the neural structure of human brain, and processes information accordingly. The performance of neural network depends on the amount of training it undergoes. Higher the volume of input signals greater will be the ability of the technique to predict accurately. Hence we factored on availing maximum possible compositions of patters that describe normal and intrusion characteristics [12]. The input signals are passed with weights that determine the importance of the signals to be processed. Furthermore these weights play a significant role in training the neural network.

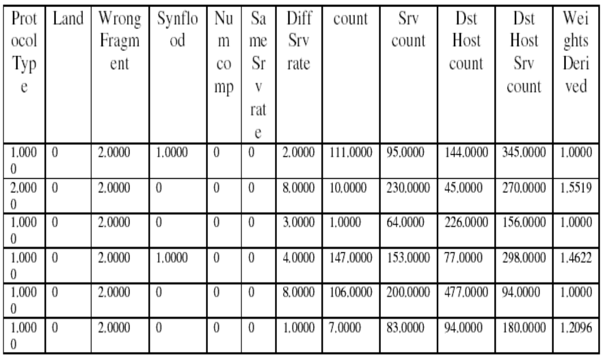
Here in our algorithm the weights assigned to the fuzzy rules generated, manipulate the way the neural network technique trains. The performance of neural network techniques is proportionate to the amount of training it receives. This training also needs to be varied so that the technique can yield substantial degree of performance even in unforeseen circumstances. The algorithm proposed takes advantage of this feature of training and offers significant alterations in the weights assigned for fuzzy rules and trains the neural network. With such a versatile knowledge gained through training, the algorithm gets equipped itself to shield the network / system in an indestructible manner.

4. Experimental Results

The following table sorts the list of parameters extracted from the DARPA datasets with the meanings they signify

Serial No.	**Name of the parameter**	**Meaning**
1	protocol_type	Type of protocol
2	land	Flag to identify whether connection is from/to the same host/port
3	wrong_fragment	Number of wrong fragments in the connection
4	synflood	Connections that have “SYN” errors
5	num_compromised	Number of compromised conditions
6	same_srv_rate	Percentage of connections to the same services
7	diff_srv_rate	Percentage of connections to the different services
8	count	Number of connections from the same source host to the same destination host
9	srv_count	Number of connections from the same source service to the same destination service
10	dst_host_count	Number of connections from the same destination host to the same source host

The values of the above fields present in each packet are temporarily stored in a vector. Using a random function all the possible combinations of these values scattered across these fields is analyzed. The algorithm appraises the values of these parameters along with the measured count of its presence. Then it calibrates the weights in accordance by dividing the counts thus derived by the total number of records present in each cluster. All these values along with their computed weights are subsequently forwarded to the Neural Network algorithm. The following table contains a sample of such inputs accumulated during experimental analysis.



Protocol Type	Land	Wrong Fragment	Synflood	Num comp	Same Srv rate	Diff Srv rate	count	Srv count	Dst Host count	Dst Host Srv count	Weights Derived
1.0000	0	2.0000	1.0000	0	0	2.0000	111.0000	95.0000	144.0000	345.0000	1.0000
2.0000	0	2.0000	0	0	0	8.0000	10.0000	230.0000	45.0000	270.0000	1.5519
1.0000	0	2.0000	0	0	0	3.0000	1.0000	64.0000	226.0000	156.0000	1.0000
1.0000	0	2.0000	1.0000	0	0	4.0000	147.0000	153.0000	77.0000	298.0000	1.4622
1.0000	0	2.0000	0	0	0	8.0000	106.0000	200.0000	477.0000	94.0000	1.0000
1.0000	0	2.0000	0	0	0	1.0000	7.0000	83.0000	94.0000	180.0000	1.2096

Neural network gains the knowledge of distinguishing the normal and abnormal packets from this training. Furthermore it renders artificial intelligence enough to capture packets that posses similar traits to that of deviant ones.

5. Conclusion

The solution crafted with an object to create a powerful intrusion detection approach proved worthwhile. Convergence of K-Means, Fuzzy, and Neural Network has helped achieve a robust architecture. The inherent deficiencies perceived in these individual techniques towards attaining an effective intrusion detection algorithm were rectified by blending them appropriately. Comprehensive analysis of the characteristics of the abnormal and even the normal packets helped recognition of their patterns and discrimination efficiently. Experimental analysis using DARPA network traffic datasets yielded significant improvement in reducing intrusion and false alarm rates.

6. References

[1] Heady R., Luger G., Maccabe A., and Servilla M. 1990. The architecture of a Network level intrusion detection system, Technical Report, CS90-20, Dept. of Computer Science, University of New Mexico, Albuquerque, NM 87131.

[2] Denning D. (1987) “An Intrusion-Detection Model,” IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, pp.222-232.

[3] Kumar S., Spafford E. H. (1994) “An Application of Pattern Matching in Intrusion Detection,” Technical Report CSD-TR-94-013. Purdue University.

[4] Ryan J., Lin M-J., Miikkulainen R. (1998) “Intrusion Detection with Neural Networks,” Advances in Neural Information Processing Systems, Vol. 10, Cambridge, MA: MIT Press.

[5] Terran lane, Carla E. Brodley, Temporal Sequence Learning and Data Reduction for anomaly Detection, Vol. 2, No. 3, August 1999, pp. 295- 331.

[6] Masayuki Murakami, Nakaji Honda. A study on the modeling ability of the IDS method: A soft computing technique using pattern-based information processing, International Journal of Approximate Reasoning, Volume 45, Issue 3 August 2007, Pages 470-487**.**

[7] Marimuthu, A. Shanmugam, A**.** Intelligent progression for anomaly intrusion detection, Applied Machine Intelligence and Informatics, 2008. SAMI 2008. 6th International Symposium, Page(s): 261-265, ISBN: 978-1-4244-2106-0, Jan 2008.

[8] Fuzzy sets, “Wikipedia”, url: http://en.wikipedia.org/wiki/Fuzzy_set

[9] Orfila, A. Carbo, J. Ribagorda, A. Fuzzy logic on decision model for IDS, Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference, Publication Date: 25-28 May 2003 Volume: 2, Page(s): 1237- 1242 vol.2, ISBN: 0-7803-7810-5.

[10] Ganesh Kumar, P. Devaraj, D. Network Intrusion Detection using Hybrid Neural Networks, Signal Processing, Communications and Networking, 2007. ICSCN '07. International Conference. Publication Date: 22-24 Feb. 2007. Page(s): 563-569, ISBN: 1-4244-0997-7.

[11] Tich Phuoc Tran; Jan, T, Boosted Modified Probabilistic Neural Network (BMPNN) for Network Intrusion Detection**,** Neural Networks, 2006. IJCNN apos;06. International Joint Conference, Volume, Issue, 0-0 0 Page(s): 2354 – 2361, Digital Object Identifier 10.1109/IJCNN.2006.247058.

[12] Sandhya Peddabachigari, Ajith Abraham, Crina Grosan, Johnson Thomas, Modeling intrusion detection system using hybrid intelligent systems, Science Direct, Journal of Network and Computer Applications 30 (2007) 114–132, June 2005.