Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 17 Issue 1 Version 1.0 Year 2017
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A New Networks Intrusion Detection Architecture based on Neural Networks

By Berlin H. Lekagning Djionang & Dr. Gilbert Tindo
University of Yaoundé I

Abstract- Networks intrusion detection systems allow to detect attacks which cannot be detected by firewalls. The false positive and false negative problem tend to make IDS inefficient. To improve those systems' performances, it is necessary to select the most relevant that will lead to characterize a normal profile or an attack. We have proposed in this paper a new intrusion detection system architecture and a scheme to flexibly select groups of attributes using neural networks in order to improve results that we have got with our architecture. The selection approach is based on a contribution criteria that we have defined in function of precision measures of type HVS (Heuristic for Variable Selection).The selected subset depends on a threshold that we make vary in function of a defined criteria. He have done a comparative study of this approach and the one without attributes selection.

Keywords: *NIDS, neural network, features selection, MLP, NSL-KDD data set.*

GJCST-E Classification: *C.2.1, C.2.2*

ANEWNETWORKSINTRUSIONDETECTIONARCHITECTUREBASEDONNEURALNETWORKS

Strictly as per the compliance and regulations of:

A New Networks Intrusion Detection Architecture based on Neural Networks

Berlin H. Lekagning Djionang [α] & Dr. Gilbert Tindo [σ]

Abstract- Networks intrusion detection systems allow to detect attacks which cannot be detected by firewalls. The false positive and false negative problem tend to make IDS inefficient. To improve those systems' performances, it is necessary to select the most relevant that will lead to characterize a normal profile or an attack. We have proposed in this paper a new intrusion detection system architecture and a scheme to flexibly select groups of attributes using neural networks in order to improve results that we have got with our architecture. The selection approach is based on a contribution criteria that we have defined in function of precision measures of type HVS (Heuristic for Variable Selection).The selected subset depends on a threshold that we make vary in function of a defined criteria. He have done a comparative study of this approach and the one without attributes selection. A comparative study has also been done with others works. The NSL-KDD dataset has been used to train, teste and evaluate our scheme. Our Works shows satisfactory results.

Keywords: NIDS, neural network, features selection, MLP, NSL-KDD data set.

I. Introduction

Interconnecting systems via computer networks has been a necessity seen the 21st century. These net works are subjects to many attacks. Intrusion detection systems are a security mechanism that allows to detect attacks which has not been identified by the firewall. An intrusion being each action that can threaten confidentiality, integrity and resources availability in an information system.

The intrusion detections systems that use neural networks as classification scheme has been widely studied by many authors [1]. Most of the solution proposed in the literature have the problem of pertinence and reliability. One of the problems major of the NIDS with neuronal networks is that the performance is governed by an only big system which takes care to detect either the types, or the categories of attacks. In this work, we have proposed a modular architecture and we have presented the efficiency. In this paper, we will explore the path of selecting attributes in order to improve the efficiency of this architecture that means to obtain a good approximation function, an acceptable false positive and negative rate and a recognition rate that is not far from the ideal one. It consists on displaying relevant attributes for each normal packet and for each type of attack.

The Learning quality of a scheme based on neural networks is linked to the quality of data that we submit to the classifier [2]. Data submitted to the classifier can influence it in many manners [3, 4]: -the recognition rate -The time required for the learning stage to obtain a satisfying recognition rate -The number of sample data necessary to obtain a satisfying recognition rate -The identification of relevant attributes - Reduce the complexity of the classifier and the execution time. Relevant attributes selection can lead to build a normal profile of a user or a particular type of attack. Input data characterization has a significant impact on many aspects of the classifier.

The follow-up of our work is organized as following: in section 2, we present the basics elements of attributes selection; in section 3, we will briefly present neural networks and their importance compared to other classifiers. In section 4 we will show some works related to attributes selection; in section 5 we will describe our attributes selection approach and algorithm, in section 6, we will present the dataset used and the preprocessing done, then in section 7 we'll present the results obtained and their analysis. We will end this work with a conclusion and prospects in section 8.

II. Attributes Selection

Relevant attributes selection is a difficult problem. Attributes selections consist on identifying a subset of attributes that allows to better the performances of detection system. It helps to remove non relevant attributes, redundant or noised ones. We will in the following subsection present the elements that help to implement an efficient selection process.

a) Basics Elements of Selection

According to [5], the main procedure follows these four steps:

a- Generation procedure: allows to explore the search space in order to find relevant subsets. [6] regroups them in three categories:- complete generation that consists on exhaustively search in the whole dataset, which is done in $O(2^N)$. – Sequential generation which consists on incrementally generate the relevant subset on the whole dataset. –Heuristic generation which is similar to the complete generation with a predefined maximum number of iterations.

The optimal subset is evaluated using an evaluation criteria [7].

b- Evaluation: It takes as input a subset of attributes and outputs a numeric value. It allows to evaluate the

Author α σ: University of Yaoundé I, Faculty of sciences, Yaoundé, Cameroon. e-mails: dberlinherve@gmail.com, gtindo@uycde.uninet.cm

examined subset. The aim of the search algorithm is to maximize the evaluation function. [5, 8] consider many types of evaluation functions: The distance measure, the information measure, the dependency measure, the classifier recognition rate, the consistency criteria, and the precision measure.

c- Stopping criteria: It allows to know when the learning algorithm should stop since the optimum number of variables is unknown in advance.

d- Validation method: allows to make sure that the selected attributes subset is valid, to determine the number of relevant attributes, to choose different parameters and to test global performances of the system [8].

b) Selection Method Based On Neural Networks

Three main approach has been proposed in the literature to implement this procedure *[4, 5]*. We have the filter approach, the wrappers approach and the embedded approach**.** The filter approach selects attributes regardless of the classifier. The wrapper approach uses the classifier to validate the subset of relevant attributes. It uses for this purpose two strategies: the for ward selection which consists to gradually add attributes and the backward selection which consists to gradually remove the attributes. The embedded approach makes attributes selection in parallel to the classification process.

III. Neural Networks

Neural networks are strongly linked networks made of elementary processors functioning in parallel and linked by weighs. These connections weighs chair the network functioning. Each elementary processor computes a unique output based on information taken as inputs. Neural networks has many advantages in implementing an intrusion detection system. They are really efficient and fast in the classification task. They are able to learn and easily identify new threats which are submitted to them. Neural networks are able to handle incomplete data, imprecise and from various sources. The natural speed of neural networks help to reduce damages when a threat is detected [10]. Neural networks usage helps to extract nonlinear relationships that exist between different fields of a packet and to timely-detect complex attacks [11]. Neural networks, after having correctly learnt, have a good generalization ability, which means that they are able to compute with precision corresponding outputs even for data which have not been learnt. The flexibility that offer neural networks is also one of the asset of intrusion detection [9].

IV. Some Works Related to Attributes Selection

Relevant variables selection help to improve the classifier efficiency. [12] are the first to use neural networks for selecting attributes with the KDD dataset. They select relevant attributes by attack categories and use only one precision criteria from [13]. [14] uses selective analysis in their work to select relevant variables. They then use this set to classify attacks. [15] Uses information gain to determine the attributes which allow to better distinguish each type of attack. [16] Proposes a combination of approaches for network intrusion detection. They use for this purpose the genetic algorithm for attributes selection and SVM (Support Vector Machine) for classification. [17] Proposes a new selection method based on the total mean of each field's class. The selected subset is evaluated using the decision tree classifier.

V. Architecture, Approach and Selection Algorithm

Attributes selection help to find out among a set of attributes, the most relevant and those which help to better the efficiency and the performance of the classifier for a given problem. Each selection depending on the system architecture, we will first present the architecture of our solution proposed in [22]. Then we will present in this section the approach that we use and the selection algorithm that we have designed.

a) Proposed Architecture

The architecture that we have used in our works is the one shown in [22],on which performances have been studied. As shown in Figure 1, it is a modular architecture organised in four stages. We have called this architecture MAMBiM: Multiple Attack Multiple Binary MLP.

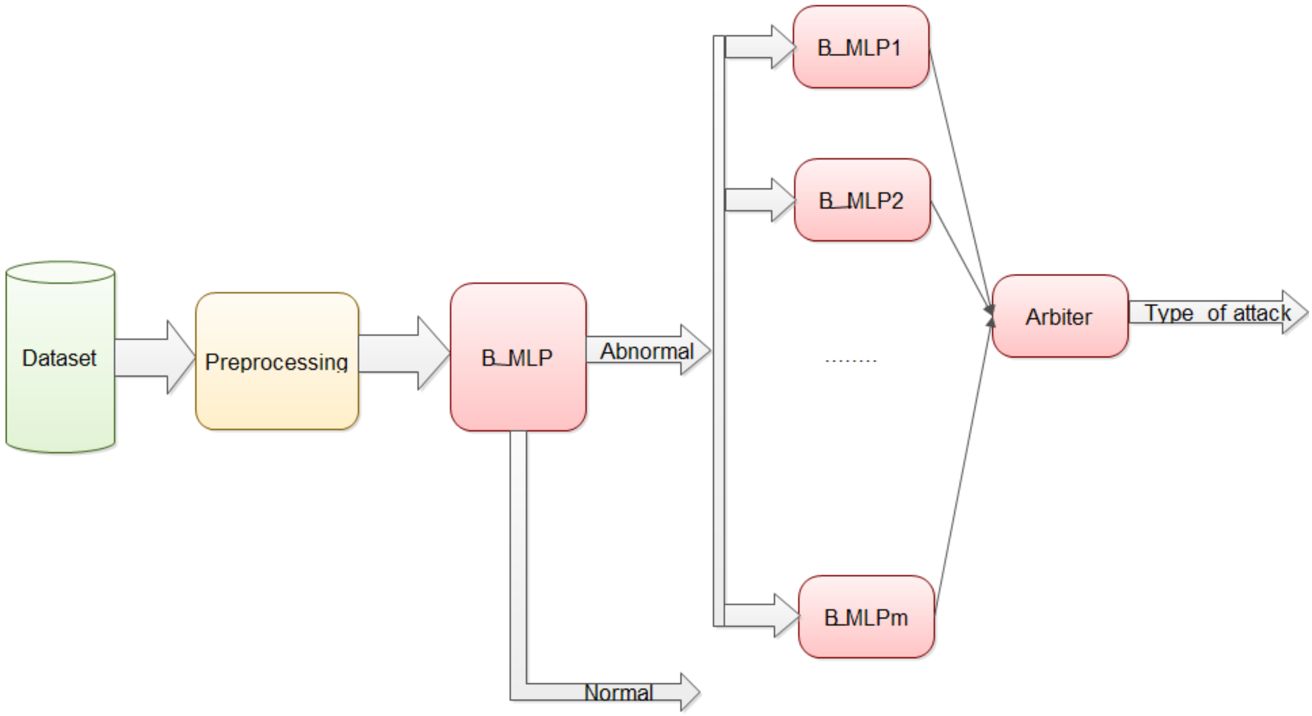

Figure 1: Four-level intrusion detection architecture (MAMBiM)

In this four-level architecture, the first level helps to preprocess data. The second one discriminate normal packets from abnormal ones. If the packet analyzed is abnormal, the nit it is thrown to other models (third level) to determine the type of attack. Element A (fourth level) in this architecture stands as a referee which will decide which type of attack it is. Each module is a neural network with one entry stage, one hidden stage and one output stage.

To better the results obtained with our architecture in [22], we have chosen the heuristic approach bas -ed on neural network to select relevant attributes.

b) Selection Approach Used

Evaluation criteria that we have used are presented in [2]. The generation procedure is a heuristic. The approach that we use is the one based on using neural model to select relevant attributes. We have proposed a relevance measure inspired from entropy. This measure is presented in (a). We will also present the measure having zero order given in [2] to evaluate the efficiency of our precision measure. This measure is described in diagram (b). The contribution formula that we propose in our work to evaluate an attribute contribution compared to the others is described in (c). Our approach implies a comparative study of the architecture performances in accordance with different precision measures chosen.

$$P_i = \sum_{j=1}^{h}\left(\left(\frac{|w_{ij}|}{\sum_{k=1}^{n}|w_{kj}|}\left|\log\left(\frac{|w_{ij}|}{\sum_{k=1}^{n}|w_{kj}|}\right)\right|\right) * \frac{|w_j|}{\sum_{l=1}^{h}|w_l|}\right)$$

(a)

$$P_i = \sum_{j=1}^{h}\left(\frac{|w_{ij}|}{\sum_{k=1}^{n}|w_{kj}|}\frac{|w_j|}{\sum_{l=1}^{h}|w_l|}\right)$$

(b)

$$C_i = \frac{P_i}{\sum_{j=1}^{n}|p_j|}$$

(c)

The measure presented by YACOUP in (b) neglect the information quantity factor contained in

$$\log\left(\frac{|w_{ij}|}{\sum_{k=1}^{n}|w_{kj}|}\right).$$

Our measure has two parts: - the part $\left(\frac{|w_{ij}|}{\sum_{k=1}^{n}|w_{kj}|}\log\left(\frac{|w_{ij}|}{\sum_{k=1}^{n}|w_{kj}|}\right)\right)$ determines the influence of input neurons weighs on the hidden layer. ; - the last part $\frac{|w_j|}{\sum_{l=1}^{h}|w_l|}$ determines the influence of output neurons on the target. P_i determines the influence of the variable *i* on the final decision.

We will then make a comparative study of performances compared to the model which has been trained by the set of attributes from the variables space. The selection approach that we will use is a wrappers approach from blocks variables downward strategy. It is illustrated in **figure 1**. And this is based on criteria (c).

c) Our Selection Algorihm

We do mention here that the error retro propagation algorithm which is used to train the neural net work.

The principle of our selection method is described in the following steps:

- Learn the network with the set of variables (of size N)from the space of variables using the errors retro propagation algorithm ;
- Evaluate the pertinence of each attribute using formulas (a) or (b) ;
- Evaluate the contribution of each variable using formula (c) ;
- Choose a contribution criteria of our choice : a threshold ϴ ;
- select the variable which satisfy the threshold ($C_i \geq \Theta$) as relevant, we obtain a set E' with size N-P, P being the number of variables that do not satisfy the condition ;
- Dynamically look for the number of neurons from hidden layer, which gives the best performance with this set of chosen variables ;
- Evaluate the network using this set and compare the performances with performances of networks with no variables selection;
- Repeat until the choice of the threshold (3) matches with the performance targeted in terms pf false positive, false negative and recognition rate.

VI. Test Dataset and Preprocessing

Since 1999, KDD Cup 99 is used as sample dataset in behavioural intrusion detection systems. Each packet from the KDD Cup 99 dataset is made of 41 fields and is labeled as a normal or an abnormal packet with types of attacks. Amidst these fields, 37 are of type numeric and 4 are of type non numeric. KDD99 combine 37 types of attacks. These attacks are subdivided in four major classes: DOS, U2R, R2L and Probes [19, 20].

- **DOS (Denial of service attacks):** they are attacks that target to threaten availability of services by overloading computers resources, servers or target networks. These attacks succeeded in networks have as consequence to freeze network traffic.
- **Probes:** attack which aims to gather information on the target that can help an attacker to trigger an attack. There exist many types of probes attacks: some abuse legitimate users and others use engineering techniques to gather information.
- **R2L (Remote to Local):** attack which aims to bypass or usurp authentication credentials to execute commands. Most of these attacks derive from social engineering [18].
- **U2R (User to Root):** This attack comes from inside. The attacker usurp the super administrator password and thus the other users' passwords. Most of these attacks come from buffer overloading caused by programming errors [19].

KDD99 dataset contains many redundant packets in training data, as in test data [20]. Redundant data are able to give more importance to a type of attack than it merits. [20] propose NSL-KDD which is an excellent dataset for comparing network IDS. Our experimentation has been done with NSL-KDD, the type of attack and the number in the training and test datasets are proposed in **table 4** in appendix. The fields in the packets are described in **table 5** in appendix.

a) Preprocessing

Pre-processing focus on non-numeric fields. Non numeric fields are: type of protocol (TCP, UDP, ICMP), type of service (AOL, auth, bgp, Z39_50), flag (OTH, REJ, RSTO, RSTOS0, RSTR, S0, S1, S2, S3, SF, SH) and the packet's class (Normal or Abnormal). For type of protocol, we assign the following numeric values: TCP=1, UDP=2 and ICMP=3. We assign 1 to normal packets and 0 to abnormal packets. For field type of service and flag, we can assign numeric values in their total number ascendant or descendant order. [21] has shown the limits of such an approach. He propose to assign random values to those fields. In our work we have assigned random values from 1 to 10 to fields of type flag, and random values from 1 to 65 to fields of type of services.

b) Normalization

It consist on transforming data to make them vary between 0 and 1, in order to make them homogeneous and thus simplify network learning. We will in this paper use the Min-Max normalization. Let be min_x and man_x respectively the minimum and the maximum of values of attribute X of value V, the normalized value is $\boldsymbol{V}' = \frac{v - min_x}{man_x - min_x}$. For each attribute of data vector, compute its normalized value and replace it with the normalized value.

VII. Experiment and Results Analysis

To evaluate our models, we will use many indicators: recognition rat (TR), false positive recognition rate (TFP), detection rate (TR) and false negative rate (TFN). This rate is computed as following:

$$\boldsymbol{TR} = \frac{NN+AA}{NN+AA+AN+NA} * \mathbf{100},$$

$$\boldsymbol{TFP} = \frac{NA}{NA+AA} * \mathbf{100},$$

Global Journal of Computer Science and Technology (E) Volume XVII Issue I Version I Year 2017

$$TFN = \frac{AN}{AN+NN} * 100,$$ with:

NN: normal packet detected as normal;

NA: normal packet detected as abnormal;

AN: abnormal packet detected as Normal;

AA: abnormal packet detected as abnormal.

For experiments, 80% of data has been used for training purposes, in which 20% are reserved for evaluation and 20% of data are used for testing. The set of data that we submit to each network is reduced compared to initial data.

a) Results analysis with a dynamic threshold

Here we present results obtained. The fields of packets from dataset are presented in appendix in table 5. This first table presents results with criteria (a).We have only presented some types of attacks. After that, we have presented the results per type of attack with our performance measure and we have compared with YACOUP measure.

Table 1: Results analysis

ATTACKS	ϴ	NV	VARIABLES SELECTED	TR%	TFP%	TFN%
Warezmaster	0	41	11111111111111111111111111111111111111111	100	0	0
	1	32	11110111111110011111111110100011110111110	100	0	0
	2	22	01110101111110001010101110000000110011110	100	0	0
	3	11	00010100001110000000001010000000000011110	100	0	0
Nmap	0	41	11111111111111111111111111111111111111111	95,9	4,25	4,78
	1	38	11111111111101110111111111111111111111100	100	0	0
portsweep	0	41	11111111111111111111111111111111111111111	99,9	0,55	0,15
	1	31	1111111111110111111111110110100000111110	98,0	4,3	0
	2	19	1111011010100010101110110000000001010100	97,5	5,3	0,4
	3	12	11100000100000101010100110000000001010000	98,0	1,8	2,08
satan	0	41	11111111111111111111111111111111111111111	96,9	4,4	2,7
	1	25	10001001011111100010011111111000100111111	95,3	6,2	3,2
	2	18	10001000011111000010000111110000100001111	91,2	10,8	7,4
	3	14	00001000011111000010000111110000100001111	90,9	11,8	7,0
pod	0	41	11111111111111111111111111111111111111111	96,5	4,4	2,4
	1	30	1100100110111111111111011111100100011111	98,8	0	2,2
	2	11	11001000000100110010000011000100100000000	100	0	0
rootkit	0	41	11111111111111111111111111111111111111111	80	33,3	0
	1	17	10000000000110110010000111111010000101011	100	0	0
	2	11	10000000000110110010000001101000000000011	80	0	25
	3					

For the attacks presented, we observe how the recognition rate gets better as we remove non relevant attributes. This allows us to present new descriptors for each type of attack. This work allows us to better the results we have presented in [22].

i. *Comparative study of our criteria with Yacoup one*

Table 2: Comparative study of our criteria with Yacoup one

Category	Type of attack	DJIONANG		YACOUP	
		Number VA	TR (%)	Number VA	TR (%)
R2L	ftp_write	39	100	37	100
	guess_passwd	31	93,02	28	93,02
	phf	40	100	34	100
	warezmaster	11	100	11	100

PROBES	ipsweep	24	99,1	24	99,1
	nmap	18	86,90	26	97,04
	portsweep	31	99,18	31	97,7
	satan	30	95,52	25	95,32
	back	41	70,52	40	68,30
	land	41	100	38	100
DOS	neptune	21	99,62	15	99,10
	pod	30	98,84	21	97,67
	smurf	41	99,7	41	99,7
	teardrop	41	99,7	41	99,7
U2R	buffer_overflow	40	84,62	30	100
	loadmodule	40	100	5	100
	perl	41	66,67	30	66,67
	rootkit	7	80	17	100
	warezclient	41	97,63	34	96,84

Taking in consideration this table, we can see that our criteria give better results compared to Yacoup criteria. In contrast, the number of variables necessary to obtain this result is broadly greater than the number of variables generated with Yacoup criteria. We have by this work displayed descriptors for each type of attack with neural network model. We notice that when the number of variables decreases in the neural network model, the learning rate also decreases for some type of attack.

ii. *Comparative study with other works*

We propose in the following table a comparative study of our work with works done by three authors on designing NIDS with explicative variables selection. Our results are presented in two columns: the first deals with a learning scheme without selection whereas the second deals with our work based selection. The non-convincing results have been better with dynamic selection. The previous table present a comparative study of the two criteria.

Table 3: Comparative study with other works

Category	Type of attack	DJIONANG		SIVA	GOLOKO
		Without selection % [22]	With Selection%	%	%
R2L	ftp_write	60	40	33,3	100
	guess_passwd	93,01	94	100	100
	imap	83,33	84	100	9,09
	multihop	33,3	66,7	22.2	0
	phf	100	100	100	100
	warezmaster	100	100	95.2	94,12
PROBES	ipsweep	99,35	100	97.1	93,93
	nmap	95,48	100	100	48,29
	portsweep	99,67	100	100	47,98
	satan	96,48	100	99.8	96,45
	back	70,52	68,30	99.4	100
	land	100	100	100	0
DOS	neptune	99,96	93,96	100	80,6
	pod	96,51	100	100	0
	smurf	99,7	99,7	100	100
	teardrop	98,96	100	66,7	100
U2R	buffer_overflow	100	100	68,2	0
	loadmodule	100	100	100	0
	perl	33,3	66,7	100	0
	rootkit	80	100	23,1	100
	warezclient	96,84	97,63	-	100

The results clearly show that our results are clearly better than works of the authors who have dealt with intrusion detection by type of attack.

VIII. Conclusion

We have in this paper, proposed a modular architecture for network intrusion systems based on neural networks and proposed an algorithm for selecting attributes that allows us to propose descriptors for each type of attack. These new descriptors have helped us to better predict different types of attack. In terms of perspectives, we plan to propose a NIDS which timely detects networks attack.

References Références Referencias

1. Berlin H Lekagning Djionang and Gilbert Tindo."Network Intrusion Detection Systems based Neural Network:A Comparative Study". International Journal of Computer Applications 157(5):42-47, January 2017
2. Philippe LERAY and Patrick GALLINARI « Feature Selection with Neural Networks" Behaviormetrika, Vol 26, pp 16-42, 1998
3. Olivier Lezoray «Segmentation d'images par morphologie mathématique et classification de données par réseaux de neurones : Application a la classification de cellules en cytologie des séreuses » THESE UNIVERSITE de CAEN/BASSE-NORMANDIE janvier 2000.
4. Saba EL FERCHICHI "sélection et extraction d'attributs pour les problèmes de classification" THESE UNIVERSITE de LILLE janvier 2013.
5. Dash, M. and Liu, H. "Feature selection for classification. Intelligent Data Analysis",1. 131 -156. (1997)
6. José Crispín HERNÁNDEZHERNÁNDEZ « Algorith - mes métaheuristiques hybrides pour la sélection de gènes et la classification de données de biopuces » THESE UNIVERSITE de ANGERS novembre 2008.
7. Saba EL FERCHICHI "sélection et extraction d'attributs pour les problèmes de classification" THESE UNIVERSITE de LILLE janvier 2013.
8. Guyon, I. and Elisseeff, A. (2003) An introduction to variable and feature selection. Journal of Machine Learning Research, 3. 1157-1182. October, Arlington, VA, pp . 443 -456. 1 998
9. James Canady "Artificial Neural Networks for Misuse Detection," Proceedings, National Information Systems Security Conference (NISSC), 98
10. G. DREYFUS "les réseaux de neurones" Mécanique Industriel et Matériaux, n51, septembre 1998
11. Vladimir Golovko, Pavel Kochurko "Intruision recognition using neural networks" International Scientific Journal of computing, 2005, vol. 4, Issue3, 37-42
12. Adel Ammar, Khaled Al-Shalfan "On Attack-Relevant Ranking of Network Features" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 11, 2015
13. MEZIANE YACOUB & YOUNES BENNAMI "Feature selection and architecture optimization in connectionist system" International journal of Neural Systems, vol 10, No 5(2000), 379-395
14. S. Siva Sathya && all "Discriminant Anlysis based feature Selection in KDD Intrusion Dataset" International Journal of Computer Application Volume 31- No. 11 october 2011
15. H. Günes Kayacık && all "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets"
16. Behrooz Mabadi Jahromy && all "A New Method for Detecting Network Intrusion by Using a combinaison of Genetic and Support Vector Machine" Journal Of Engineering and Applied Science 11 (4) 810-815, 2016
17. H. S Chae, B. O. Jo, S. H. Choi, and T. K. Park, "Feature Selection for Intrusion Detection using NS L-KDD," Recent Advances in Computer Science, 2013, 184-187.
18. Srinivas Mukkamala && all "Intrusion detection using an ensemble of intelligent paradigms", Journal Network and Computer Applications 28 (2005), 167-182
19. Matthew Vincent Mahoney "A Machine Learning Approach to Detecting Attacks by Identifying Anomalies in Network Traffic ", these of Florida Institute of Technology, May 2003
20. Mahbod Tavallaee && all "A Detailed Analysis of the KDD CUP 99 Data Set" Proceding of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Application (CISDA 2009)
21. Aslihan Ozkaya & Bekir Karlik "Protocole Type Based Intrusion Detection Using RBF Neural Network" International Journal of Artificial Intelligence and Expert Systems(IJAE), volume(3): Issue(4):2012
22. BHL DJIONANG, G TINDO. "Towards A New Architecture of Detecting Networks Intrusion Based on Neural Network." International Journal of Computer Networks and Communications Security 5, no. 1 (2017): 7-18.

Appendix

Categories of Attacks In Nls-Kdd99 Dataset

Table 4: Type of Attack Per Category

Category	Type of attack	Training	Test	Category	Type of attack	Training	Test
Normal	Normal	67 343	9711	DOS	neptune	41214	4657
R2L	ftp_write	8	3		pod	201	41
	guess_passwd	53	1231		processtable	0	685
	httptunnel	0	133		smurf	2646	665
	imap	11	1		teardrop	892	12
	multihop	7	18		udpstorm	0	2
	named	0	17	U2R	buffer_overflow	30	20
	phf	4	2		loadmodule	9	2
	sendmail	0	14		perl	3	2
	snmpgetattack	0	178		ps	0	15
	snmpguess	0	331		rootkit	10	13
	warezmaster	20	944		sqlattack	0	2
	worm	0	2		xterm	0	13
	xlock	0	9				
	xsnoop	0	4				
Probes	ipsweep	3599	141				
	mscan	0	996				
	nmap	1493	13				
	portsweep	2931	157				
	saint	0	319				
	satan	3633	735				
DOS	apache2	0	734				
	back	956	359				
	land	18	7				
	mailbomb	0	293				

Different Attributes of Nsl-Kdd Dataset

Table 5 : List of Attributes with Description And Type

N°	Attribute	Description	Type
1	Duration	Duration of connection	cont
2	Protocol type	Connection protocol (tcp ou udp)	disc
3	Service	Destination service (telnet, ftp)	disc
4	Flag	Status flag of connection	disc
5	Source bytes	Byte send from source to destination	cont
6	Destination bytes	Bytes send from destination to source	cont
7	Land	1 if connection is from/to the same host/port; 0 otherwise	disc
8	Wrong fragment	Number of wrong fragments	cont
9	Urgent	Number of urgent packets	cont
10	Hot	Number of "hot" indicators	cont
11	failed logins	Number of failed logins	cont
12	Logged in	1 if successfully logged in; 0 otherwise	disc
13	Number of “compromised” conditions	Number of "compromised" conditions	cont
14	Root shell	1 if root shell is obtained; 0 otherwise	cont
15	“Su root” command attempted	1 if "su root" command attempted; 0 otherwise	cont
16	Number of “root” accesses	Number of "root" accesses	cont
17	Number of file creations	Number of file creation operations	cont

18	Number of shells prompts	Number of shell prompts	cont
19	Number of operations on access files	Number of operations on access control files	cont
20	Number of outbound commands	Number of outbound commands in an ftp session	cont
21	Is host login	1 if the login belongs to the "hot" list; 0 otherwise	disc
22	Is guest login	1 if the login is a "guest" login; otherwise	disc
23	Count	Number of connections to the same host as the current connection in the past two seconds	cont
24	Service count	Number of connections to the same service as the current connection in the past two seconds	cont
25	Syn error rate	% of connections that have "SYN" errors	cont
26	Service Syn error rate	% of connections that have "SYN" errors	cont
27	Rej error rate	% of connections that have "REJ" errors	cont
28	Service Rej error rate	% of connections that have "REJ" errors	cont
29	Same service rate	% of connections to the same service	cont
30	Different service rate	% of connections to different services	cont
31	Service different host rate	% of connections to different hosts	cont
32	Same destination host count	count of connections having the same destination host	cont
33	Same destination host and service count	count of connections having the same destination host and using the same service	cont
34	Same destination host and service rate	% of connections having the same destination host and using the same service	cont
35	Different services on current host	% of different services on the current host	cont
36	Connect to current host with same source error	% of connections to the current host having the same src port	cont
37	Connect to same service from diff. host	% of connections to the same service coming from different hosts	cont
38	Connect to current host with S0 error	% of connections to the current host that have an S0 error	cont
39	Connect to current host and specified service that have an S0 error	% of connections to the current host and specified service that have an S0 error	continu
40	Connect to current host with RST error	% of connections to the current host that have an RST error	continu
41	Connect to current host and specified service with RST error	% of connections to the current host and specified service that have an RST error	continu