Journal of Theoretical and Applied Information Technology
31st December 2014. Vol.70 No.3
© 2005 - 2014 JATIT & LLS. All rights reserved·

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPLICATION OF FEAST (FEATURE SELECTION TOOLBOX) IN IDS (INTRUSION DETECTION SYSTEMS)

[1] MENDOZA PALECHOR FABIO, [2] DE LA HOZ CORREA EDUARDO, [3] DE LA HOZ MANOTAS ALEXIS

[1]Asstt Prof., Department of Systems Engineering, Universidad de la Costa, Colombia
[2, 3]Assoc. Prof., Department of Systems Engineering, Universidad de la Costa, Colombia
E-mail: [1]fmendoza1@cuc.edu.co , [2] edelahoz6@cuc.edu.co, [3] adelahoz6@cuc.edu.co

ABSTRACT

Security in computer networks has become a critical point for many organizations, but keeping data integrity demands time and large economic investments, in consequence there has been several solution approaches between hardware and software but sometimes these has become inefficient for attacks detection. This paper presents research results obtained implementing algorithms from FEAST, a Matlab Toolbox with the purpose of selecting the method with better precision results for different attacks detection using the least number of features. The Data Set NSL-KDD was taken as reference. The Relief method obtained the best precision levels for attack detection: 86.20 %(NORMAL), 85.71% (DOS), 88.42% (PROBE), 93.11 %(U2R), 90.07(R2L), which makes it a promising technique for features selection in data network intrusions.

Keywords: *Feature Selection Toolbox* (*FEAST), Data-Set, Security, Attacks, Networks*

1. INTRODUCTION

Computer networks were originally designed for a limited number of users, nowadays they have become a necessity for homes and also small, medium and large organizations. Inadequate designs in computer network structures have generated security breaches, which takes us to need new strategies that allow the identification of unauthorized access to the network to keep the integrity, confidentiality and availability of the information being transferred in them.

The aplication of different artificial intelligence techniques (genetic algorithsm, decision tree, artificial neural networks, among others) have managed to put into practice the same effectiveness for the detection of attacks in computer networks [1, 2, 3, 4].

The application of different techniques or methods of disciplines as data mining and machine learning is widely used today for analysis of big data sets. In this research, the features selection technique "feast" was applied to choose the best features for big data volumes.

The use of data sets composed by normal and anomaly traffic captures from real scenarios is highly important, which have made that datasets as nsl-kdd, kdd-cup 99 or darpa 98, has been used to the study and development of intrusion detection systems as shown in [5, 6, 7, 8]. The use of this datasets implies the analysis of the data that they englobe, and it's mandatory a data preparation phase to be able to make the correct features selection.

2. BACKGROUND WORK

Attacks to information systems keep growing every day, given the new facilities provided by several hackers and crackers sites, and the incremental knowledge of computer tools and weaknesses like the case of Windows in its different versions, port vulnerabilities, viruses, backdoors and troyan horses [9].

So far, several studies have focused in attack detection in computer networks. These studies have applied a number of techniques to achieve a positive percentage, using various datasets. These are some examples:

- [10] Shows a dimension reduction method PCA for the data preparation and then implements a neural classifier achieving 99.3% as result in attack identification. The entry data is a vector of 419 variables reduced to only 20 variables.

- [11] Implements genetic algorithms for normal traffic pattern recognition obtaining promising results given coincidences for the dataset in 80%.
- [12] Designs an intruder detection method base don svm (Support Vectorial Machines) looking for attacks identification in computer networks obtaining 94% of precision.
- [13] Builds an instruction detection method using the CSI-KNN K-closing neighbors algorithm. The algorithm analyze the different features of data of the network using two measures: strangeness and isolation. Based on these measures, a correlation unit increase intrusion alerts with the trust estimation associated.
- [14] Use the data from DARPA 1998 to study the intruder detection systems, comparing performance of Robust Support Vectorial Machines (RVSMs) against Support Vectorial Conventional machines and classifier of the nearest neighbor. The results indicate the RSVMs superiority in terms of high precision in intrusion detection and few false positives, also in their capacity for generalization in the presence of noise and working time.

3. FEATURES SELECTION

The features selection is a term used usually in data mining to describe the tools and available techniques to reduce the entries to an appropriate size for analysis and processing.

The features selection implies the cardinality reduction, which is the imposition of an arbitrary or predefined limit in the numbers of attributes that are considered to create a model, and also selecting the attributes, where the analyst or the modeling tool must choose or dismiss them according to their usefulness to the analysis.

The ability to apply the features selection is essential for an efficient analysis, since the datasets usually contain much more information than the required for the model generation, which can degrade the quality of patters detected by the following reasons:

- Some columns are noisy or redundant. This noise makes more difficult relevant pattern detection based on data.
- To detect good quality patterns, most data mining algorithms require a training dataset bigger in a multidimensional dataset. Nevertheless, in some data mining applications, we have a lack of training data.

A typical features selection process takes 4 steps [15]:

Subset generation, is the search mechanism to produce subset of candidate features to be evaluated.

The complete search guarantees obtaining the optimal subset, without having to search all the possible subsets (2n) from all n features, which is an exhaustive search [16].

The sequential search generate subsets in a direct way, starting with an empty subset, then adding relevant features progressively, or starting with the whole set and dismissing irrelevant features [17].

The random search generate subsets in random manner, then increase or decrease features randomly to obtain the next subset to be evaluated.
Subset evaluation, measures the optimization degree of the subset generated in the previous step according a learning problem, in this case classification.

The evaluation criteria filter is independent from the learning algorithm (e.g. Neural networks, support vector machines, etc.). Involving evaluation, depends on the algorithm used. This one can have a bigger computational cost than the ones of filter model.

Stop criteria, determines when a features selection process must stop. Usually this happens when certain parameter level is reached, the full search is completed or an optimal features subset has been found.

Results validation, is the evaluation of the features selection model with real data and to check if the performance shows clearly the result of the experimentation.

4. FEATURE SELECTION TOOLBOX (FEAST)

Feast provides the implementation of features selection algorithms based on common information, and an implementation of relief. All functions expect discrete entries (with exception of relief, which doesn't depends on mitoolbox) and return the indexes of the selected function. All the feast code is available under bsd license.

Feast contains implementation methods as: mim, mrmr, mifs, cmim, jmi, disr, cife, icap, condred, cmi, relief, fcbf, betagamma [18].

5. DATA SOURCE

Nsl-kdd is a dataset to resolve some of the inherent problems of kdd'99. Altough, this new version of kdd still have some of the previous problems, and it doesn't represent perfectly existing networks, due to the lack of public datasets for ids based on network, still can be applied as a reference dataset established to help researchers to compare the different methods for intruder detection.

In other way, the number of register of nsl-kdd and test teams is reasonable. This advantage makes it accessible for executing experiments about the whole dataset without the need of selecting randomly a small section. In consequence, the results of evaluations made by the different research will be consistent and comparable.

Since 1999, the dataset kdd'99 has been the most used for evaluation in anomaly detection methods. This dataset was prepared by stolfo [19] and built over the database captured in darpa'98. Darpa'98 is a dataset product from 7 weeks of network traffic, with 5 million connections, each of them with 100 bytes. The two weeks of test data have around 2 million of connection registers.

The dataset kdd is composed of 4.900.000 connection vectors approximately, each of them with 41 features and tagged as normal or attack, defining the kind of attack. The simulated attacks fall in the following four categories [20]:

• Denial of service (dos): these attacks try to stop a network, machine or process or deny the use of their resources or services to authorized access. There are two types of dos attacks:

Operating system attacks, the ones which try to exploit failures and can be avoided applying the right patches.

Network attacks that exploit limitations inherent to the protocols and network infrastructure. There are several types of dos, some of them like "mailbomb", "neptune" or "smurf" abuse of legitimate devices. Others like "teardrop", create malformed packages that confuse the tcp/ip protocols of the target machine and this one will try the reconstruction of the packages. Others like "apache2" or "back" take advantage of network errors.

• Probing: this type of attacks scan networks trying to identify valid ip address and collect information about them (offered services, operating systems). Usually, this information provides the attacker a list of potential vulnerabilities that can be used to carry attacks to the services and target machines.
These attacks are the most frequent, and usually are precursors of other ones. An attacker with a map of the machines and available services in a network can use this information to find weak spots of it. Some of this analysis tools "satan", "saint", "mscan" allow that a newbie hacker can easily check hundreds or thousands of machines in a network.

• Remote to local (r2l): this kind of attack happens when an attacker doesn't have any account to a machine, obtains access (as user or root) to it. In most attacks r2l, the attacker access the information system through internet.
There are several ways than an attacker can achieve his objective. Some attacks exploit the buffer overflow caused by network server software "imap", "named", "sendmail". The attacks of "ftp", "write", "xsnoop" and "guest" try to exploit the weakness or wrong configuration of security policies of the system.

The "xlock" attack uses social engineering to succeed, the attacker must surpass to human operators that provide their passwords to screensavers that are really trojan horses.
User to root (u2r): this type of attack happens when an attacker that has access to an account in a information system is able to elevate their privileges exploiting existing vulnerabilities, a hole in the operating system or an installed software.

There are several types of u2r attacks, the most common is the "buffer_overflow" attack, that is produced when a software copies a big amount of data in a static memory buffer without checking the size of it, which produce an overflow. The overflowed data is stored in the overflow stack of the system, covering the next instructions to be executed. Through the manipulation of the stack, an attacker can provoke the execution of code in the operating system that would help him to achieve his objectives. Another kind of u2r attacks exploit the programs that provide information about their execution environment, a good example of this kind of attack is “lodamodule”.

Other kind of u2r attacks exploit the programs that have a wrong temporary files management. Some u2r attacks exploit the vulnerability due to exploitable competitive conditions during the execution of one, two or more programs simultaneously. Nevertheless, a controlled programming could eliminate all the vulnerabilities, most unix and windows versions still have them present day.

6. METHODOLOGY

This work is based on training a dataset taken from NSL-KDD to select the best method in the FEAST. The steps used by this research and data capture follow:

1. Download the NSL-KDD dataset, which contains the data necessary for the development of this research.

2. Use of the Matlab Software R2013a, to implement the appropriate code for the execution of the different methods of FEAST.

3. Data Distribution Selection for training and classification. For this we took 20% of training data with 100% of test data; 100% of training data with 100% of test data.

4. Once selected the data distribution we proceeded to execute the code to obtain the results of each method of FEAST, each of them was tested with 5, 10, 15, 20, 25, 30 and 35 features to analyze the features rank which provides the best precision percentages.

5. Comparative Graphs and tables building for better observation of the results found.

7. RESULTS

The classification process was developed using Matlab R2013a. The initial data provided by the NSL-KDD dataset follows:

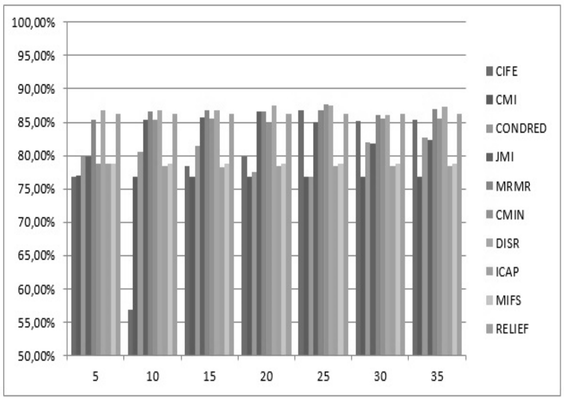
- KDDTrain+20Percent;41 attributes; 25192 records.
- KDDTrain +100Percent;41 attributes; 125973 registers.
- KDDTest + 41 attributes; 22544 registers

We applied 10 FEAST methods to be able to compare them and identify which of them shows a better feature selection, given the different accuracy levels when submitted to execution with the same dataset. The compared methods were: mrmr, mifs, cmim, jmi, disr, cife, icap, condred, cmi, and relief.

Next, we show in Table N° 1, the accuracy results obtained by each FEAST method crossed reference with the type of attack “NORMAL, DOS, PROBE, U2R, R2L”.

Table No 1. *Accuracy percentages FEAST methods Normal Connections*

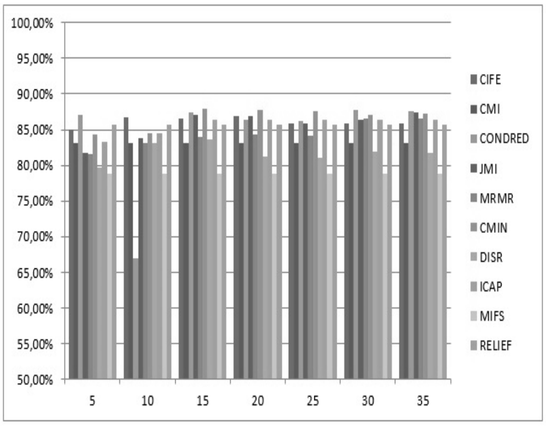
	5	10	15	20	25	30	35
CIFE	76,90	56,92	78,48	79,90	86,75	85,23	85,37
CMI	77,03	76,76	76,76	76,76	76,76	76,76	76,76
CONDRED	79,92	80,57	81,48	77,54	76,85	81,96	82,64
JMI	79,92	85,42	85,70	86,60	85,07	81,72	82,41
MRMR	85,42	86,64	86,71	86,69	86,86	86,06	86,96
CMIN	78,83	85,39	85,47	84,81	87,73	85,48	85,46
DISR	86,76	86,82	86,82	87,51	87,50	86,12	87,30
ICAP	78,83	78,46	78,25	78,39	78,39	78,39	78,39
MIFS	78,72	78,71	78,71	78,71	78,71	78,71	78,69
RELIEF	86,20	86,20	86,20	86,2	86,20	86,20	86,20



Graph No 1. Accuracy percentages FEAST methods Normal Connections

Table No 2. *Accuracy percentages FEAST methods DoS Attacks*

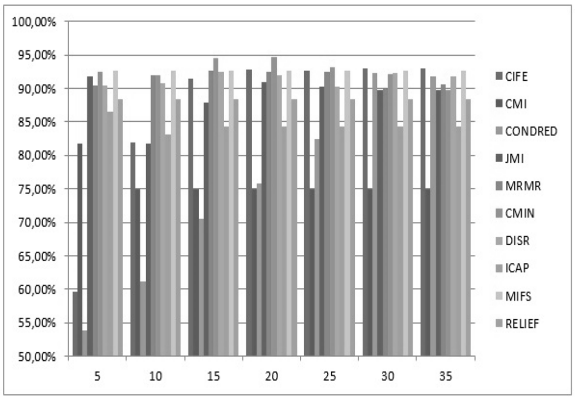
	5	10	15	20	25	30	35
CIFE	85,04	86,65	86,62	86,83	85,90	85,87	85,87
CMI	83,13	83,13	83,13	83,13	83,13	83,13	83,13
CONDRED	87,06	66,92	87,34	86,39	86,28	87,74	87,68
JMI	81,72	83,81	87,02	86,89	85,94	86,43	87,42
MRMR	81,50	83,06	83,98	84,27	84,20	86,60	86,52
CMIN	84,31	84,52	87,98	87,69	87,57	87,15	87,26
DISR	79,72	83,06	83,57	81,15	80,99	81,84	81,73
ICAP	83,23	84,54	86,43	86,43	86,43	86,43	86,43
MIFS	78,80	78,81	78,81	78,81	78,81	78,81	78,81
RELIEF	85,71	85,71	85,71	85,71	85,71	85,71	85,71



Graph No 2. *Accuracy percentages FEAST methods DoS Attacks*

Table No 3. *Accuracy percentages FEAST methods PROBE Attacks*

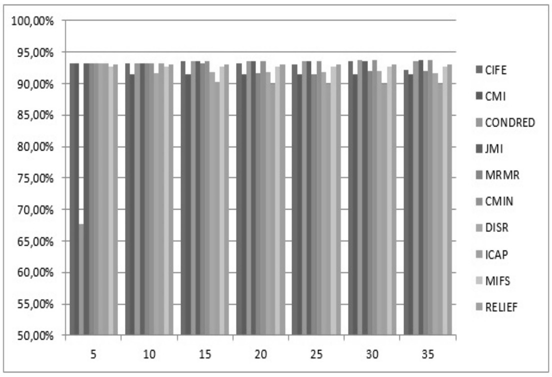
	5	10	15	20	25	30	35
CIFE	59,68	81,88	91,47	92,85	92,71	92,91	93,04
CMI	81,79	74,89	74,89	74,89	74,89	74,89	74,89
CONDRED	53,76	61,15	70,54	75,77	82,39	92,31	91,77
JMI	91,84	81,72	87,85	90,95	90,19	89,76	89,74
MRMR	90,44	91,90	92,66	92,56	92,43	90,12	90,55
CMIN	92,46	91,89	94,46	94,66	93,10	92,13	89,81
DISR	90,44	90,74	92,56	91,91	90,26	92,27	91,86
ICAP	86,60	83,03	84,32	84,32	84,32	84,32	84,32
MIFS	92,69	92,69	92,69	92,69	92,69	92,69	92,69
RELIEF	88,42	88,42	88,42	88,42	88,42	88,42	88,42



Graph No 3. Accuracy percentages FEAST methods PROBE Attacks

Table No 4. *Accuracy percentages FEAST methods U2R Attacks*

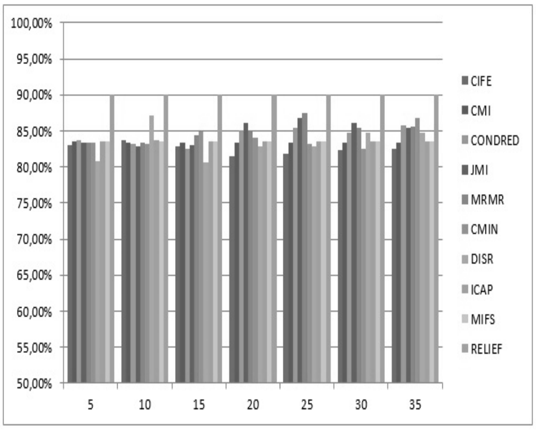
	5	10	15	20	25	30	35
CIFE	93,20	93,18	93,63	93,19	93,00	93,48	92,09
CMI	93,19	91,50	91,50	91,50	91,50	91,50	91,50
CONDRED	67,65	93,19	93,49	93,49	93,59	93,65	93,47
JMI	93,21	93,19	93,53	93,53	93,54	93,57	93,66
MRMR	93,21	93,26	93,16	91,69	91,47	91,90	91,91
CMIN	93,23	93,22	93,49	93,50	93,62	93,70	93,80
DISR	93,22	91,64	91,73	91,79	91,89	91,90	91,71
ICAP	93,23	93,18	90,21	89,97	89,97	89,97	89,97
MIFS	92,66	92,66	92,66	92,66	92,66	92,66	92,66
RELIEF	93,11	93,11	93,11	93,11	93,11	93,11	93,11



Graph No 4. Accuracy percentages FEAST methods U2R Attacks

Table No 5. *Accuracy percentages FEAST methods R2L Attacks*

	5	10	15	20	25	30	35
CIFE	83,02	83,64	82,83	81,52	81,87	82,31	82,54
CMI	83,46	83,39	83,39	83,39	83,39	83,39	83,39
CONDRED	83,65	83,13	82,54	85,01	85,41	84,74	85,73
JMI	83,38	82,78	82,94	86,06	86,75	86,11	85,36
MRMR	83,38	83,43	84,40	84,90	87,53	85,47	85,65
CMIN	83,38	83,21	84,84	84,03	83,26	82,43	86,79
DISR	80,87	87,11	80,65	82,77	82,76	84,79	84,75
ICAP	83,46	83,75	83,61	83,61	83,61	83,61	83,61
MIFS	83,54	83,54	83,54	83,54	83,54	83,54	83,54
RELIEF	90,07	90,07	90,07	90,07	90,07	90,07	90,07



Graph No 5. Accuracy percentages FEAST methods R2L Attacks

8. CONCLUSION

The FEAST methods allow data analysis to big scale to achieve better results related to the attack identification in computer networks after testing methods as mrmr, mifs, cmim, jmi, disr, cife, icap, condred, cmi, and relief.

After full validation of the methods using the data set NSL-KDD, the RELIEF method shows a precision of 86.20% (Normal Behavior), 85.71% (DOS Attack), 88.42% (PROBE Attack), 93.11% (U2R attack), and 90.07% (R2L attack) which clearly indicates promising results as a features selection technique, being tested with five different features of the dataset. The CONCRED method shows the lowest precision results of 87.06% (Normal Behavior), 53.76% (PROBE attack), 67.65% (U2R attack), and 83.65% (R2l attack). The remaining methods show acceptable results for data analysis which makes them apt to be used as feature selection methods for future developments.

REFRENCES:

[1] M. Crosbie and G. Spafford., "Applying genetic programming to intrusion detection. " in AAAI Fall Symposium on Genetic Pro- gramming, 1995.

[2] R. Gong, M. Zulkernine, and P. Abolmaesmumi, "A software implementation of a genetic algorithm based approach to network intrusion detection.," in Sixth Internatio- nal Conference on Software Engineering, Artificial Intelligence, Networking and Para- llel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNDP/SWAN'05), vol. 0, pp. 246–253, 2005.

[3] W. Li, "A genetic approach to network intrusion detection," tech. rep., SANS Institute, 2003.

[4] C. Sinclair, P. Lyn, and S. Matzer, "An application of machine learning to network intrusion detection.," in 15th Annual Compu- ter Security Applications Conference, 1999.

[5] Herrera, D., Carvajal, Helber., IMPLEMENTACIÓN DE UNA RED NEURONAL PARA LA DETECCIÓN DE INTRUSIONES EN UNA RED TCP/IP, Revista Ingenierías USBMed, paginas 45-48, 2010.

[6] Kayacık, G., Zincir, N., Heywood, M., Selecting Features for Intrusion Detection: A Feature

Relevance Analysis on KDD 99 Intrusion Detection Datasets , Dalhousie University, Faculty of Computer Science

[7] P. Ananthi y P. Balasubramanie, «A Fuzzy Neural Network And Multiple Kernel Fuzzy C-Means Algorithm For Secured Intrusion Detection System,» Journal of Theoretical and Applied Information Technology (JATIT), pp. 206-217.

[8] A. Falke, V. Fulsoundar, R. Pawase, S. Wale y S. Ghule, «Network Intrusion Detection System using Fuzzy Logic,» nternational Journal Of Scientific Research And Education, pp. 626-635.

[9] Castillo, R., Deteccion de Intrusos Mediante Tecnicas de Mineria de Datos, Departamento de Sistemas e Informatica, Universidad Autonoma de Colombia.

[10] Lorenzo, I., Macia, F., Mora, F., Gil, J., Marcos, J., Modelo Eficiente y Escalable para la Deteccion de Intrusos en Red, Departamento de Tecnologia y Computacion, Universidad de Alicante.

[11] Catania, C., Garcia, C., 2008, Reconocimiento de Patrones en el Trafico de Red Basado en Algoritmos Geneticos, Revista Iberoamericana de Inteligencia Artificial, Vol 12, 65-75.

[12] Xiaoqing, G., Hebin, G., Luyi, C., Network Intrusion Detection Method Based on Agent and SVM, Beijing Vocational College of Electronic Science Beijing 100026,P.R. China.

[13] Kuang, L., Zulkernine, M., An Anomaly Intrusion Detection Method Using the CSI-KNN Algorithm School of Computing Queen's University Kingston, Canada.

[14] W. Hu, Y. Liao, and V. Vemuri. Robust Support Vector Machines for Anomaly Detection in Computer Security. Proc. International Conference on Machine Learning and Applications, pages 23–24, 2003.

[15] Oporto, Sl., Aquino, I., Chavez, J., Perez, C., Comparacion de Cuatro Tecnicas de Selección de Caracteristicas Envolventes usando Neuronales, Arboles de Decisión, Maquinas de Vector de Soporte y Clasificador Bayesiano.

[16] D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

[17] H. Liu and H. Motorola. Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academy, (1998).

[18] http://www.cs.man.ac.uk/~gbrown/fstoolbox/ (07/11/2013)

[19] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Costbased modeling for fraud and intrusion detection: Results from the jam project," discex, vol. 02, p. 1130, 2000.

[20] Tribak, H., Febrero 2012, Análisis Estadístico de Distintas Técnicas de Inteligencia Artificial en Detección de Intrusos. Tesis Doctoral