Artículo de investigación

Provides a hybrid approach to increase the accuracy of intrusion detection in cloud computing and the generation of false alarms

Disposición de un enfoque híbrido para aumentar la precisión de la detección de intrusos en la computación en la nube y la generación de falsas alarmas.

Fornecimento de uma abordagem híbrida para aumentar a precisão da detecção de intrusos na computação em nuvem e a geração de alarmes falsos.

Recibido: 9 de abril de 2017. Aceptado: 30 de mayo de 2017

Escrito por: Mohammad Hossein Ameri[53]

Abstract

As cloud computing services are presented via internet, the security and privacy are the key issues these services encounter. The open and distributive (decentralized) structure of cloud computing has changed this kind of computing into targets for hackers, cyber attackers and intruders. The relevant studies conducted by the International Data Research Institute reveal that security is the biggest challenge for cloud computing.

For efficiency and more effectiveness of intrusion detection systems, they should take detection in real-time and online. Intrusion detection systems (IDS) using signature-based detection techniques (like snort) are careful and real-time intrusion systems, which can detect known attacks immediately and activate security mechanisms according to known attack patterns and known attach databank. However, these systems practically lose their efficiency against unknown attacks. Hence, this study has tried to present a hybrid method to enhance intrusion detection accuracy in cloud computing and the amount of production of false warnings in them.

Key words: cloud computing, intrusion, intrusion detection, warning management

Resumen

Dado que los servicios de computación en la nube se presentan a través de Internet, la seguridad y la privacidad son los problemas clave que enfrentan estos servicios. La estructura abierta y distributiva (descentralizada) de la computación en la nube ha transformado este tipo de informática en objetivos para piratas informáticos, atacantes cibernéticos e intrusos. Los estudios pertinentes realizados por el Instituto Internacional de Investigación de Datos revelan que la seguridad es el mayor desafío para la computación en la nube.

Para lograr una mayor eficiencia y efectividad de los sistemas de detección de intrusos, deben detectarse en tiempo real y en línea. Los sistemas de detección de intrusiones (IDS) que usan técnicas de detección basadas en firmas (como snort) son sistemas de intrusión cuidadosos y en tiempo real, que pueden detectar ataques conocidos inmediatamente y activar mecanismos de seguridad de acuerdo con patrones de ataque conocidos y banco de datos adjunto conocido. Sin embargo, estos sistemas prácticamente pierden su eficacia contra ataques desconocidos. Por lo tanto, este estudio ha intentado presentar un método híbrido para mejorar la precisión de la detección de intrusos en la computación en la nube y la cantidad de producción de advertencias falsas en ellos.

Palabras clave: computación en la nube, intrusión, detección de intrusos, gestión de advertencias.

[53] mha.matrix@yandex.com
Kourosh Nemati :PHD
knms81@gmail.com. Islamic Azad University Nour Branch- Department of computer science

Resumo

Como os serviços de computação em nuvem são apresentados via Internet, a segurança e a privacidade são os principais problemas que esses serviços encontram. A estrutura aberta e distributiva (descentralizada) da computação em nuvem transformou esse tipo de computação em alvos para hackers, invasores cibernéticos e intrusos. Os estudos relevantes conduzidos pelo International Data Research Institute revelam que a segurança é o maior desafio para a computação em nuvem.

Para eficiência e mais eficácia dos sistemas de detecção de intrusão, eles devem fazer a detecção em tempo real e on-line. Sistemas de detecção de invasão (IDS) usando técnicas de detecção baseadas em assinatura (como snort) são sistemas de intrusão cuidadosos e em tempo real, que podem detectar ataques conhecidos imediatamente e ativar mecanismos de segurança de acordo com padrões de ataque conhecidos e bancos de dados conhecidos. No entanto, esses sistemas praticamente perdem sua eficiência contra ataques desconhecidos. Portanto, este estudo tentou apresentar um método híbrido para melhorar a precisão da detecção de intrusão na computação em nuvem e a quantidade de produção de alertas falsos neles.

Palavras-chave: computação em nuvem, intrusão, detecção de intrusão, gerenciamento de alertas.

Introduction

Nowadays, use of information communication technology (ICT) is being deformed and a new concept called cloud computing is being evolved. No exact, certain and same definition is available for cloud computing and the definition is being changed and developed constantly. Cloud computing is computing model based on large computer networks like internet providing a novel and modern model to supply, consume and deliver IT services including hardware, software, data and other computing sharing sources via internet. In a study, Moody et al (2012) has defined cloud computing as a computing model providing applied sources and programs as service and via internet for the users. In another work, Moody (Moody et al, 2013) has mentioned that cloud computing is aimed in providing an easy and on-demand network access to a set of configurable computing sources (like networks, servers, storage memories and application).

National Institute of Standards and Technology (NIST) (Bahador, M and Keshtkar M.M, 2017) has introduced cloud computing through counting its 5 fundamental features (wideband network, rapid flexibility, measurable service, on-demand service supply) and 3 service providing models (software as a service SaaS, platform as a service PaaS and infrastructure as a service IaaS). For more efficiency of intrusion detection systems, these systems should be online detection systems; i.e. these systems should take detection in real time. Offline detection systems or post hoc attack systems can be only used for the footprint of an attack; although they can never detect an occurring attack and neutralize it. If attacks are all in kind of known attacks, the high capability and efficiency of the intrusion detection techniques such as signature-based intrusion detection techniques in these systems can be used. As these systems use signature databank of known attacks to detect the intrusive attacks, they are too fast and can hence provide the capability of online detection for these systems. However, when the attacks are unknown, these mechanisms lose their efficiency in practice. Efficiency of intrusion detection systems is depended on true and false warnings produced by these systems against the attacks. The cause of the importance is that protection systems take measure based on same analyses, detections and warnings against the received traffic. The measure can lead to blockage of the special traffic. Now, if the warning is sent falsely, it can have important negative effects on data availability and reliability of system. Woos and Zhang (2009) divided the computing evolution process from terminals (dummy terminal) and mainframe to the grid computing and cloud computing to 6 phases. In phase 1, lots of users used to apply several terminals with high computing capability and storage of mainframes through sharing method. In phase 2, stand-alone PCs gained power to an extent that they could meet main needs of users. In phase 3, PCs, laptops and servers were linked through the local networks to share the sources through this and enhance efficiency. In phase 4, various local networks were linked to other local networks to use the remote programs and sources and formed a worldwide network like

internet. In phase 5, grid computing shares the storage ability through a distributed computing system. In phase 6, cloud computing shares more sources through the internet and using a simple, flexible and scalable method. The main objective of this study is helping enhancement of security of components and infrastructures of cloud computing in all organizations now using this type of computing or those who are going to use that in the way of their progress perspective. For example, at the current world that the e-commerce and attraction of investment through virtual world is important, more security making is equal to gaining more trust of customers and ultimately, attracting high amount of investment. Same issue can be generalized in other way to educational and academic environments and libraries (e.g. in field of electronic education), hospitals and medical departments (e.g. distance surgical operations) and other environments using this technology.

Intrusion and intrusion detection

The National Institute for Standards and Technology has defined intrusion as an effort to endanger security policies (privacy, generality and availability) or bypassing computer and network security mechanisms. Moreover, the institute has defined intrusion detection as the process of monitoring computer and network system events and analysis of intrusion signals. Moreover, the institute believes that intrusion detection systems are software/hardware systems for automation of intrusion detection process (Keshtkar M.M , 2013).

Keshtkar, and Ghazanfari (2013) have classified intrusion detection systems based on detection method, the deployment of these systems inside the cloud computing systems and due to their structure and their reaction to attacks as it is illustrated in figure 1. In figure 1, it can be observed that Shanmugam has classified intrusion detection systems to two groups based on type of reaction to attacks:

- Active intrusion detection systems

Passive intrusion detection systems

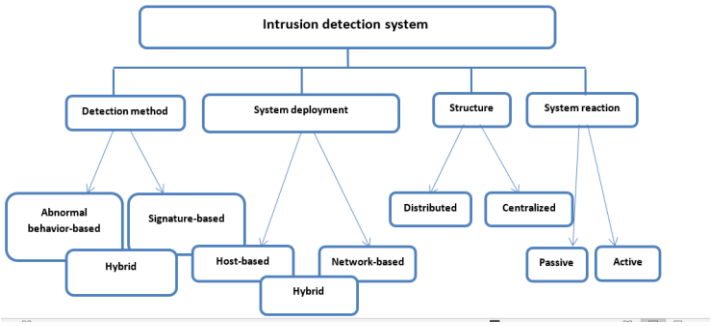

Figure 1: classification of intrusion detection systems in view of Shanmugam

Moreover, Patel (Keshtkar, 2013) has provided new classification for intrusion detection systems, based on which the systems are divided to 2 groups in terms of intrusion encountering time:

- Real time IDS: intrusion detection systems showing reaction to intrusions in real time and online mode.
- Offline IDS: intrusion detection systems showing reaction to intrusion in offline and non-real time mode. In this regard, Goverison (Keshtkar, 2013) has compared several classification algorithms in IDS. The results obtained from the said study are analyzed in table 1. According to obtained results from this study, the computing time of rule-based algorithms is shorter than neural network-based algorithms and its false classification percentage is also lower than neural network-based algorithms. In other words, based on the results of the study, rule-based algorithms for classification in IDS are more efficient than neural network algorithms for classification.

Table 1: the results of comparing accuracy and time of rule-based and neural network-based classification algorithms for intrusion detection (Goverison)

	Rule-based algorithm	Neural network-based algorithm
Percentage of truly classified samples	99.9%	93.9%`
Percentage of falsely classified samples	0.1%	6.1%
Computing time (sec)	239.64	467.42

Moody et al (2012) presented a network-based IDS using serial combination of snort and tree. The author used snort to detect known

attacks and used tree to classify the unknown patterns to reduce production arte of false warning and to improve and enhance detection accuracy. The author used intrusion systems on processing servers in the proposed framework, so that internal attacks can be also detected in addition to external attacks.

Proposed method

In this section, a method is presented to enhance accuracy of decision making on the packets received from the network and to reduce the production rate of false warnings. To this end, the study tends to use combination of two clustering and classification algorithms simultaneously to take benefit of advantages of both methods to enhance the accuracy.

Here, C4.5 algorithm (Roji Roy, 2002) as the developed version of decision tree algorithm is used for classification of received packets and Learning Vector Quntization (LVQ) algorithm is used for clustering. General steps of the proposed method are as follows:

- ***Making cluster using Learning Vector Quntization (LVQ) algorithm***

LVQ algorithm (Arbib, 2003) is in fact an adaptive classification method and although it is a supervised version of data mining algorithm, it can be used as non-supervisory clustering algorithm for the data (Antón Chávez, 2017 and Castillo and Melin, 2003). In fact, the algorithm can be the developed version of Self-Organized Mapping Algorithm with the difference that this algorithm lacks topological structure. Figure 2 has illustrated an example of architecture of network of this algorithm. As it is obvious in the figure, the algorithm has divided the 2-D input space (X1 and X2) to 4 clusters (Castillo and Melin, 2003).

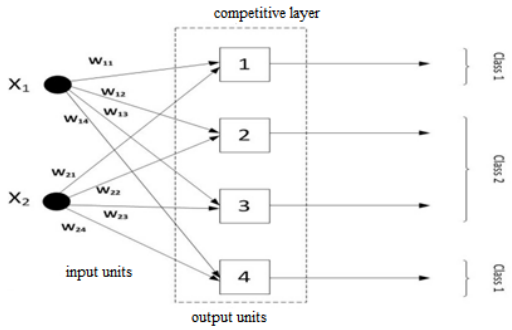

Figure 2: an example of LVQ

LVQ algorithm includes 2 general steps. In first step, a non-supervisory clustering method is used regardless of practical data class to determine the position of center of clusters. In second step, the practical data class is used to optimize output clusters and to reduce number of samples divided falsely (Castillo and Melin, 2003). LVQ algorithm is illustrated in figure 3.

Hypotheses:

- M number of clusters
- Input vector (n: number of input dimensions) $x = (x_1, x_2, x_3, \ldots, x_n)$
- Weighted vector for j output unit $w*j = (w1j, w2j, \ldots, wnj)$
- A: learning rate

Algorithm

(1) Define the number of clusters M

(2) Initializ M prototype vectors (weight vectors) (centroids): $w_{*1}, \ldots, w_{*m}$

(3) Initialize learning rate (α)

(4) While (stopping criterion is satisfied)

(4-1) Randomly pick an input x

(4-2) For j = 1 TO n

(4-2-1) Determine Winner node k by finding prototype vector that satisfies $| w_{*k} - x | <= | w_{*j} - x |$

(4-2-2) Update only the Winner prototype vector weights according to:

$w_{*k}(new) = w_{*k}(old) + \alpha (x - w_{*k}(old))$

Figure 3: LVQ clustering algorithm

- ***Making C4.5 tree classification*** .The C4.5 tree making algorithm makes ate the first a decision tree with starting from practice set D including practice samples (Roji Roy, 2002). The decision tree (as a type of greedy algorithm) is made using downward, recursive and divide & conquerer method. At the first, all practice samples of set D are in root of the tree. Then, a feature (F with f_1, f_2, f_3,…f_n values) is selected from the features of the practice samples and are divided to different groups (D1, D2,…Dn) based on same practice sample features. Same process is also applied in recursive form on each distribution cluster (Di) (Han and Camber, 2006). All samples in subset of Di have same value in fi feature.

The selective features, based on which distribution is taken, are selected based on heuristic or statistical (like information gain) methods. The stop state of this algorithm happens when (Han and Camber, 2006):

1. All remained samples are belonged to same class (meaning that is all remained samples are normal Di or all of them are intrusion)
2. No other feature is remained, based on which classification is taken
3. No other sample is remained.

ID3 algorithm has used information gain in the process of feature selection and the selected feature can be the feature with highest information gain (Han and Camber, 2006). However, the C4.5 algorithm used gain ratio to make decision node, so that the selected feature is the feature with highest gain ratio. C4.5 algorithm code network (Roji Roy, 2002) is illustrated in figure 4.

```
                                    FormTree(T)
(1) Compute_Class_Freq(T);
(2) If  OneClass or  FewCases
                    Return leaf;
Create_Decision_Node( N);
(3) ForEach Attribute A
ComputeGain(A);
(4) N.test = Attribute with Best Gain;
(5) N.test is continuous
Find_Threshold();
```

Figure 4: C4.5 tree construction code

Information gain of a feature refers to reduced entropy obtained by means of separation of the data through same feature.

Entropy refers to purity (impurity) of a set (before separation) made based on equation I (Han and Camber, 2006).

$$Entropy(D) = -\sum_{i=1}^{n\_class} P_i \times \log_2(P_i) \quad (1)$$

Where;

n-class: number of classes of classification (in this problem includes two normal and intrusion classes)

Pi: the packet weight weighbridge belonged to class Ci inside the set D to overall D set; more simply, Pi is the probability function of placement in class Ci made by eq.2:

$$P_i = \frac{|Freq(C_i,D)|}{|D|} \quad (2)$$

The expected entropy after separation of data based on selected feature is obtained as follows (Han and Camber, 2006):

$$Entropy_i(D) = \sum_{j=1}^{s} \left(\frac{|D_j|}{|D|}\right) \times Entropy(D_j) \quad (3)$$

Where;

S: number of different values of F feature; i.e. number of branches caused by analysis of F feature on D dataset in decision tree

Dj: subset of D, in which all members have j value per F feature

Now, the information gain of F feature can be obtained as follows (Han and Camber, 2006):

$$Gain(F) = Entropy(D) - Entropyf(D) \quad (4)$$

The information gain can go towards features with higher values using the above formula. To meet the problem, C4.5 algorithm can normalize information gain. To this end, the obtained information gain (eq.4) is divided to overall subset entropies (eq.5) as it is presented in eq.6.

$$SplitEntropy_f(D) = -\sum_{j=1}^{s} \left(\frac{|D_j|}{|D|}\right) \times \log_2\left(\frac{|D_j|}{|D|}\right) \quad (5)$$

$$Gain\_Ratio(F) = \frac{Gain(F)}{SplitEntropy(F)} \quad (6)$$

When the computing gain ratio was calculated per each feature, the feature with highest gain ratio is selected as selected feature for the D set branch (Gonzáles Llontop anf Otero Gonzáles, 2017 and Han and Camber, 2006). As the made decision tree may be large and encounter overfitting problem, C4.5 algorithm uses the Reduced Error Pruning technique to prune nodes not causing enhancement of accuracy of classification (ineffective in enhancement of classification accuracy) to some extent of confidence.

Overfitting happens when there are lots of branches on the decision tree and can also lead to abnormal behavior in case of existence of noise. Moreover, this problem can decrease accuracy of algorithm on unknown samples (Han and Camber, 2006).

- ***Receiving packets from the network and delivery to snort*** .As the location and position of grid IDS is on the processing servers, the packets should be received from both internal network (local or private virtual network) and external network (internet) and be delivered to snort.

- ***Analysis of packets received by snort*** .Snort receives the packets and its detection motor analyzes its content to find the link and relevance to predefined patterns. If the packet is not among attacks based on snort patterns, the packet is delivered to modulation of intrusion detection based on abnormal behavior. If the packet is among known attacks and is detected by the snort, the packet is discarded and a warning is produced.

- ***Received packet clustering.***The packet is delivered to the clustering agent and is placed in a cluster based on LVQ algorithm.

- ***Received packet classification*** .The packets of each cluster are delivered to the decision tree separately and the class of packet is specified at the end. If the packet is in normal class, it would be allowed to pass. If it is in the intrusion class, the production step (warning production and packet discard) would be the next step.

- ***Warning production and packet discarding*** .In this step, warning is produced, packet is discarded and if the warning application has reached from the modulation of intrusion detection based on abnormal behavior, the attack packet is registered in the attack registration central base. The said steps are iterated per each packet entered to the network and separate the normal packets from attacks.

Simulation of proposed method and evaluation results

To implement this method, cloud computing environment of Eucalyptus open source installed on Ubuntu operating system is used and the Weka component is also used as data mining component (for clustering and classification) and practice dataset and NSL-KDD test are also used.

Eucalyptus

Eucalyptus (Eucalyptus.com) is a software framework for cloud computing providing and implementing IaaS. These systems enable users to implement and control all types of virtual machines deployed in all physical sources (Normie et al, 2009).

Weka

Weka is a set of machine learning algorithms for data mining. Weka includes tools for preprocessing, classification, estimation of standard deviation, clustering, associative rules and visualization of packets. The main parts of Weka include explorer, experimenter, knowledge flow, simple command line interface and Java interface.

KDDCup'99 dataset

It is a dataset provided by Stolfo et al and based on the data provided by DARPA'98 IDS evaluation program and has been widely accepted by scholars in field of intrusion detection as a criterion dataset. The dataset is among rare datasets used generally for studying detection systems of network-based abnormal behavior (Tavallaei, 2009).

The whole KDDCup practice dataset contains 41 features classified in two classes of good (normal) and bad (aattack) (Tavallaei, 2009). The bad class itself is divided to 4 groups as follows:

1. No-service attacks
2. User to root attacks (unauthorized access to root scores)
3. R2L (remote to local) attack
4. Probing attack (like port explore)

KDD'99 features are classified in 3 groups (Kayachik et al, 2005):

1. Basic features: packet headers
2. Content features: domain data, which are usually used to evaluate the TCP packets. In this group, one can refer to features such as number of failed trying to sign in to a user account.
3. Time-based traffic features: these features include characteristics reaching effective level in certain time period (2sec):

1-3- similar service: it analyzes only those services providing one type of service in recent 2-sec period on the existing link

2-3- similar host: considers just services providing service only for a special server in recent period on existing link

1. Traffic features: these features emphasize number of links created to a special server in the history of links (e.g. per 100 links)

The KDD'99 practice dataset includes 22 types of attack and test data includes 17 new types of attack in addition to them and includes typically attacks not considered in practice data.

NSL-KDD dataset

NSL-KDD dataset is the advanced version of KDD'99 intrusion detection dataset presented by Tavallaei et al (2009), in which some defects of KDD'99 are met. The most underlying limitation of KDD'99 dataset is existence of lots of redundant data, so that 78% of practice data (table 2) and 75% of test data (table 3) were repetitive. Same issue has caused orientation of educational and practice algorithms towards repetitive records and this can prevent detection of attacks rarely happened.

Table 2: redundant records in KDD'99 practice dataset

	Main records	Different records	Reduction rate
Normal	972781	812814	16.44%
Attack	3925650	262178	93.32%
Total	4898431	1074992	78.05%

Table 3: redundant records in KDD'99 test dataset

	Main records	Different records	Reduction rate
Normal	60591	47911	20.92%
Attack	250436	29378	88.26%
Total	311027	77289	75.15%

The NSL-KDD dataset includes 125974 practice data and 22544 test data including 25 types of traffic (24 attack types with normal traffic) (table 4).

Table 4: information of NSL-KDD dataset

Dataset	Number of practice data	Number of test data	Number of features	Number of class types
NSL-KDD	125974	22544	42	25 types (24 attack types + 1 normal traffic)

Description of criteria and evaluation method

Percentage of samples classified properly can show the method accuracy. To this end, the confusion matrix (prediction/reality matrix) is formed at the first (Chandulikar and Nandavadkar, 2012) (table 5) according to the real-time class of packets and the class predicted for them.

In this case, the sample space is:

- A: number of negative true predictions, which have been negative in reality
- B: number of false positives, which have been negative in reality
- C: number of false negatives, which have been positive in reality
- D: number of true positives, which have been also positive in reality

Table 5: confusion matrix

		Predicted status	
		Negative	Positive
Real-time status	Negative	A	B
	Positive	C	D

Now, some criteria can be provided for evaluation (Chandulikar and Nandavadkar, 2012):

Table 6: evaluation criteria

Number	Criteria	Equations
1	Classification accuracy (Acc): true predictions to overall sample space ratio	$Acc = \frac{A+D}{\mathrm{A+B+C+D}}$
2	True positive rate (TP): true positive prediction to overall true positive ratio	$TP = \frac{D}{\mathrm{C+D}}$
3	False positive rate (FP): the false positive prediction to overall real negative ratio	$FP = \frac{B}{\mathrm{A+B}}$
4	True negative rate (TP): negative prediction to overall real negative ratio	$\mathrm{TN} = \frac{A}{A+B}$
5	False negative rate (FN): false negative prediction to overall real positive ratio	$FN = \frac{C}{\mathrm{C+D}}$
6	Precision rate: true positive prediction	$Precision = \frac{D}{\mathrm{B+D}}$

7	to overall positive prediction ratio Recall, sensitivity rate: true positive prediction to overall true positive and false negative predictions	$Recall = \frac{D}{C + D}$

The confusion matrix can be used to evaluate the intrusion detection method as it is presented in table 7 and all above presented equations can be used to this end. It could be mentioned that the parameter is same true positive rate parameter:

Table 7: confusion matrix for intrusion detection

		Predicted status	
		Normal	Intrusion
Packet real status	Normal	A	B
	Intrusion	C	D

Method results

To implement the proposed method, for intrusion detection of abnormal behavior, NSL-KDD dataset and Weka tool (with features mentioned in table 8) are used. At the first, practice data is delivered to LVQ algorithm. The algorithm divided the data to two clusters (table 8). The class of the data should not be noted in clustering.

Table 8: practice data clustering

Total number of practice data	Number of data in cluster 1	Number of data in cluster 2
125973	24039	101934
	19%	81%

Afterwards, each cluster was independently delivered to the classifier as practice data to train and make decision tree based on C4.5 algorithm. Hence, two clustering-classification models were produced.

In next step, the test data was delivered to the clustering agent as input. The clustering agent divided data to 2 clusters (table 9).

Table 9: test data clustering

Total number of test data	Number of data in cluster 1	Number of data in cluster 2
125973	8562	13982
	38%	62%

Each cluster was separately delivered to the classifier and the results in tables 10 and 11 are obtained in comparison with findings of Moody.

Table 10: confusion matrix of cluster 1

		Predicted class	
		Normal	Intrusion
Real class	Normal	2094	19
	Intrusion	28	6421

Table 11: confusion matrix of cluster 2

Predicted class		Predicted class	
		Normal	Intrusion
Real class	Normal	7553	45
	Intrusion	46	6338

Table 12: overall confusion matrix

Predicted class		Predicted class	
		Normal	Intrusion
Real class	Normal	9647	64
	Intrusion	74	12759

Table 13: results of proposed method efficiency (%) (part 1)

Dataset	True positive rate (TPR)	False positive rate (FPR)	True negative rate (TNR)	False negative rate (FNR)	Number of data
NSL-KDD (cluster 1)	99.56	0.89	99.1	0.43	8562
NSL-KDD (cluster 2)	99.27	0.59	99.4	0.72	13982
Overall weighted mean	99.38	0.7	99.29	0.61	22544
results of proposed method of Moody	76.80	4.81	95.15	23.13	22544

Table 14: results of proposed method efficiency (%) (part 2)

Dataset	Precision	Recall	F measure	Acc	Number of data
NSL-KDD (cluster 1)	98.68	99.56	99.9	99.45	8562
NSL-KDD (cluster 2)	99.39	99.27	99.6	99.34	13982
Overall weighted mean	99.12	99.38	99.71	99.38	22544
results of proposed method of Moody	95.45	76.8	85.11	84.31	22544

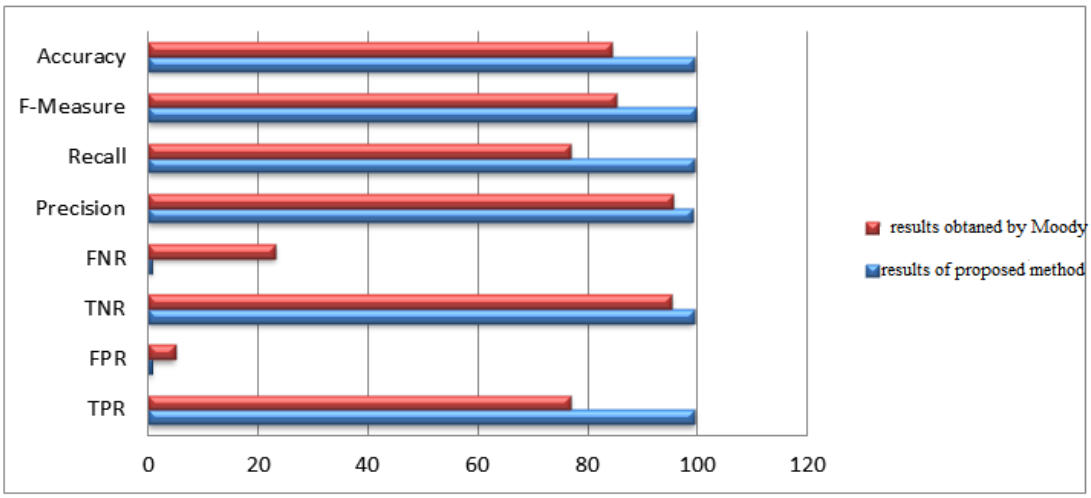

Figure 4: diagram of comparing results of proposed method in this study and proposed method of Moody

Conclusion

The efficiency of intrusion detection systems (IDS) is depended on the rate of true and false warning rate produced by these systems against intrusions; because the security mechanisms can be activated and make decision for the received packets based on same warnings. Therefore, false warning production can block and discard the normal traffic and ultimately, can reduce availability and reliability and reduce overall system efficiency.

In a recent study conducted in IDS in cloud computing, in the intrusion detection component, same signature-based (Snort) IDS is used serially and simultaneously to detect known attacks and an intrusion detection system based on abnormal behavior is also used to detect unknown attacks. In the component of abnormal behavior-based intrusion detection, decision tree is used to make decision on input packets. Hence, of the input packet was a known intrusion, it could be immediately detected by the Snort and security mechanisms used to be activated. If the packet was among unknown attacks, it used to enter to phase2 of intrusion detection and the packet used to be delivered to behavior-based intrusion component to make decision on its normal or intrusive nature.

In this study, to enhance the efficiency of applied IDS, in the component of intrusion detection based on abnormal behavior, a

clustering and a classification algorithm were used simultaneously to reduce the production arte of false warnings and to enhance intrusion detection accuracy.

References

Arbib, Michael A. The Handbook of Brain Theory & Neural Networks. MIT Press, 2003.

Archer, Jerry, and (Others). Security Guidance for Critical Areas of Focus in Cloud Computing V 3.0. CSA (Cloud Security Alliance), 2011.

Antón Chávez, A.D.P (2017). Influencia de la noticia en la imagen corporativa de una municipalidad desde la percepción del ciudadano. Opción, Año 33, No. 84 (2017): 90-119

Castillo, Oscar, and Patricia Melin. Soft Computing and Fractal Theory for Intelligent Manufacturing. springer, 2003.

Gowrison, and (Others). "Minimal complexity attack classification intrusion detection system." Elsevier, Applied Soft Computing 2013, 2013: 921-927.

Han, Jiawei, and Micheline Kamber. Data Mining: Concepts and Techniques. 2006.

Keshtkar, M. (2013) Research Article Simulation of Thermo-Hydraulic Behavior of a Lid-Driven Cavity Considering Gas Radiation Effect and a Heat Generation Zone, International Journal of Engineering & Technology, 1(1), 8-23.

Keshtkar M. M. (2013) Numerical Simulation of Radiative Heat Transfer in a Boiler Furnace Contained with a Non-Gray Gas, International Journal of Engineering & Technology, 1(3), 137-148.

Keshtkar M. M., Ghazanfari M. (2017). Numerical Investigation of Fluid Flow and Heat Transfer Inside a 2D Enclosure with Three Hot Obstacles on the Ramp under the Influence of a Magnetic Field, Engineering, Technology & Applied Science Research, 7 (3), 1647-1657.

Bahador M., Keshtkar, M. M., (2017). Reviewing and modeling the optimal output velocity of slot linear diffusers to reduce air contamination in the surgical site of operating rooms, INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY, 17 (8), 82-89.

Modi, Chiragh N, and (Others). "A novel framework for intrusion detection in cloud." ACM, Proceedings of the Fifth International Conference on Security of Information and Networks. ACM, 2012. 67-74.

S. M. Seyedhosseini, M. J. Esfahani, M. Ghaffari: A novel hybrid algorithm based on a harmony search and artificial bee colony for solving a portfolio optimization problem using a mean-semi variance approach. J Central South University 23 (2016), No. 1,181–188.

G Soleimani, M Amiri, SM Khatami, MJ Isfahani : Using S Technology, in the Automotive Industry, with the Approach of Its Implementation in Commercial Vehicles . Industrial Engineering & Management Systems.2016; 15(4) : pp.290-297

Gonzáles Llontop, R & Otero Gonzáles, C (2017). Imaginarios sociales en estudiantes de educación sobre la calidad de la formación investigativa . Opción, Año 33, No. 84 (2017): 759-790.

Modi, Chiragh N, and (Others). "A survey of intrusion detection techniques in Cloud." Journal of Network and Computer Applications, 2013: 42-57.

Modi, Chiragh N, and (Others). "Integrating Signature Apriori based Network Intrusion Detection System (NIDS) in Cloud Computing." Elsevier, Procedia Technology, 2nd International Conference on Communication, Computing & Security (ICCCS-2012). Elsevier, 2012. 905-912.

Patel, Ahmed, and (Others). "An intrusion detection and prevention system in cloud computing: A systematic review." Journal of Network and Computer Applications, 2013: 25-41.

Ruggieri, Salvatore. "Efficient C4.5 (classification algorithm)." IEEE Transactions,