ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

Packet Features Extractor for Network Security Systems: Design and Implementation

Abdullah A. Mohamed; Dia M. Ali
Communication Department-Collages of Electronics Engineering - Mosul university-Iraq

Abstract*—The Network Systems (NS) depend essentially on the packet features (header fields and payload) in their work that include packet analyses and filtering. Routers, bridges and network security systems need to extract the features of the packet to do their jobs for that the Packet Features Extractor (PFE) is one of the most important parts in the on-line NS and it is the absent step that through it the off-line NS can be developed to on-line systems. In this paper, a PFE is designed to develop an off-line Intrusion Detection System (IDS) to be an on-line system. The designing was based on extract 12 NSL-KDD features from the packet. The 12 features are the same features that the off-line IDS was designed based on it. The PFE drove the on-line IDS with same result that was the off-line IDS giving it with average detection rate 94.9%. The PFE with the new on-line IDS are implemented on Real Time Operation System (RTOS) that was created for this issue. The implementation of the system gave a high performance results and throughput reached 100Mbps that limited by the Network Interface Card (NIC) and it is thought that it can be over that if a gigabit NIC is used. It is surely that the throughput is limited by the NIC but not by the system performance. Four models are implemented and the results show that the average of the throughput saturated with convergence values for the four deferent models that simpler than the PFE and the values reach the end of the NIC limit which about 100Mbps. The system task (PFE over TCP communication) take about (0.65 micro sec) as average Task Execution Time (TET). In the rest of the paper, a study for the NSL- features reduction is produced.

***Index Terms*—Packet Features Extractor (PFE); Intrusion Detection System (IDS); Real Time Operation System (RTOS).**

I. INTRODUCTION

Today, Computer's networks became the essential factor in most of life's fields. The Data are transported in the network as frames or in general, it is named packets. The packet represents the base that all the network systems depend on it. Routers, Bridges, Ant viruses, Intrusion detection systems (IDS), Intrusion Prevention Systems (IPS) and Firewalls (FW), all they take the packet as the basic unit for their analyses and their results [1] [2]. Network Security Systems (NSS) are the most popular and important Network Systems that depend on the packet's contents on its work. The packet content is consisting of header and payload. The NSS pick up some of packet contents called "features". The features represent the header fields and some specific points of the payload. Header fields used to detect the external attacks and the internal attacks (from authorized user) but the payload features used to detect the internal attacks only. IDS and IPS designed essentially to detect the internal attacks and the external attacks for that, they make their decision according to the packet features that are represented by the header fields and the payload [3]. FW detects the external attacks only, so it focuses on the header fields [4]. IDS with its both types fig.(1) is a suitable example to show the benefit of the features extractor sub-system. The online system need to deal with real network so it sniffs the packets then extracts the features that it needs. After that, it pass the features to the system core to analyses. The offline system deals with a stored data that represent the features [5]. It selects the needed features according to the attacks that the system designed to detect it. From the previous, it clear that the offline system has not any important or benefit as an applicable system unless it connected to a features extractor. The features extractor gives the life to the network offline systems to be a useful systems and valid for Appling in the real networks.

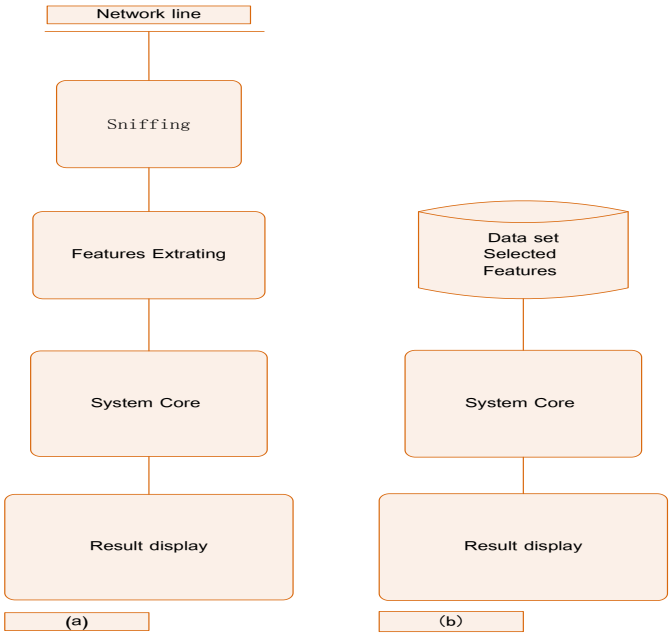


Fig. 1 IDS Architecture: (a) online IDS, (b) offline IDS

II. DATASET AND FEATURE EXTRACTING

To make the system efficient and accurate, it must be designed on specific factors. Extracting features is one of these factors, for that, it must select the effected features that have important effect on the detection operation [6]. Several foundations produced a dataset showed the effect features to detect the attack types.

A. KDD-99 and NSL-KDD Dataset:

KDD-99 is one of these dataset. It is the mostly and widely

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

used data set for the anomaly detection. It is built based on the data captured in DARPA'98 that criticized by McHugh [7], for that some of the existing problems in DARPA'98 remain in KDD-CUP 99. One of the most important deficiencies in the KDD-99 is the huge number of redundant records. Analyzing KDD-99 train and test sets discovered that about 78% of the training set and 75% of testing set are duplicated. The researchers found two important issues that highly effect on the performance of evaluated systems that result a very poor evaluation of anomaly detection approaches [8], [9]. To solve this problem, they have proposed a new dataset called "NSL-KDD". NSL-KDD solved the inherent problems of KDD-99 [10]. It consists of selected records of the complete KDD-99 dataset [8].

The following are the advantages of NSL-KDD over the original KDD-99 dataset [8]:

- It does not include redundant records in the train set, so the classifiers will not be biased towards records that are more frequent.
- The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD-99 dataset. As a result, the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques.
- The numbers of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

NSL-KDD consist of approximately 4,900,000 records each one contains 41 features. The record labeled as either normal or attack type [8]. It represent attack case and gives packet's summary of the attack. Each attack has related features as shown in table1, for that, many researchers had deep studies to specifying that relations [10], [11], [12]. For IDS, not all the 41 features are useful. Some of the features are irrelevant and redundant that results lengthy detection process and degrades the performance of the IDS [11], [10]. The table shows that for DOS, R2L and U2R attacks, 11 features are enough to achieve the IDS detection [12]. The researchers suggested many methods for features selection and reductions, all these researches involve about enhance the process time, performance and the detection rate.

Table 1the most relevant feature for each attack type and normal [12]

Attack	Most relevant features	Feature Name	Variations	Dependency ratio	Class
Back	5	source bytes	66,64,60	0.9708	DOS
Land	7	land	2	0.9999	DOS
neptune	5	source bytes	0	0.9328	DOS
Pod	8	wrong fragment	1	0.9853	DOS
Smurf	5	source bytes	39	0.7731	DOS
teardrop	8	wrong fragment	2	0.9913	DOS
Satan	30	diff srv rate	30	0.7648	PROBE
ipsweep	36	dst host name src port rate	13,14,15,17	0.8282	PROBE
Nmap	5	source bytes	4	0.6448	PROBE
portsweep	28	srv error rate	9	0.8057	PROBE
normal	29	same srv rate	28	0.8871	NORMAL
guess_passwd	11	failed login	1	0.9622	R2L
ftp_write	23	count	1	0.7897	R2L
Imap	3	service	60	0.9980	R2L
Phf	6	destination bytes	28	0.9976	R2L
multihop	23	count	1	0.7898	R2L
warezmaster	6	destination bytes	33	0.7500	R2L
warezclient	3	service	13	0.6658	R2L
Spy	39	dst host srv serror rate	8	0.9997	R2L
buffer_overflow	3	service	6	0.6965	U2R
loadmodule	36	dst host name srcport rate	29	0.6279	U2R
Perl	14	root shell	1	0.9994	U2R
rootkit	24	srv count	1	0.7269	U2R

B. The effect of Reduction on the Process time and detection accuracy

The 41 features are reduced to 4 features by using three type of Artificial Neural Networks (ANN) [13] and the results show high reduction in process time and good accuracy in the detection fig (2). It shows some deviation in the detection rate with little records but it very acceptable for 4 features only. Other researchers used 8 features [8]. Their results showed that 80.4% data reduction and approximately 35%-40% reduction in training time and 75%-80% reduction in testing time fig.(3). It also showed more accuracy in detection rate even with a little records fig.4, it clear that it more approach from the target with less deviation than the 4 features dependency [13]. Good results showed with 11 features [14], [12]. Other results was better in the false alarm with use 13 and 15 features [15], [16] respectively.

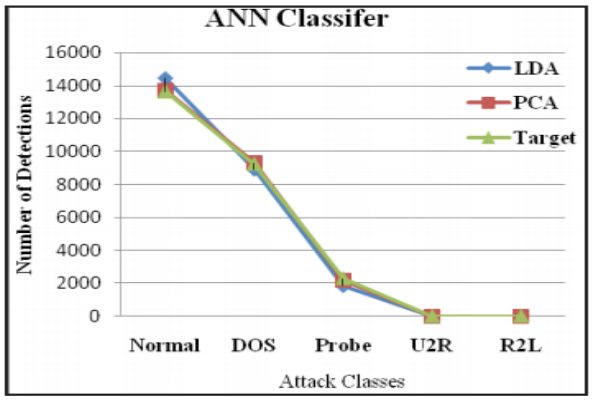
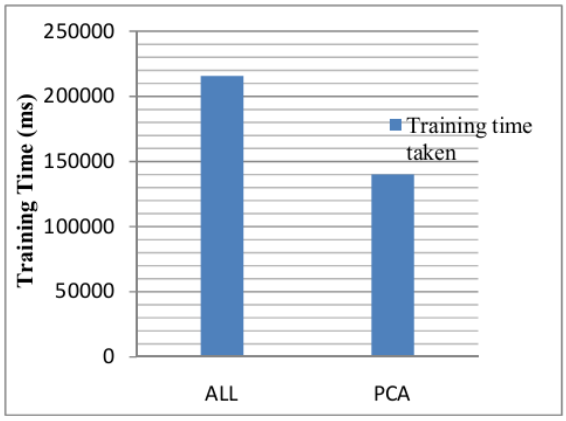
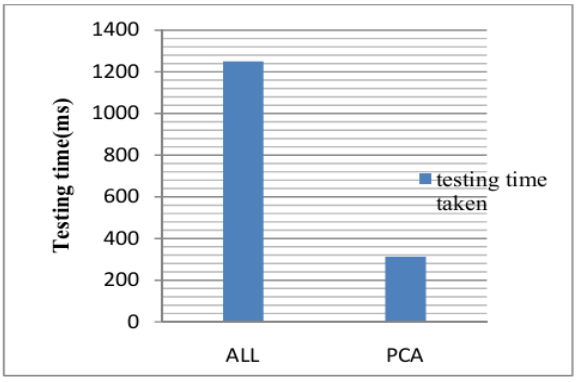


Fig. 2 Comparison of Number of Detections with LDA and PCA for 4 features [13]



(a)



(b)

Fig. 3 Time Taken before and after Reduction [8]: (a) Training Time (b) Testing Time.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

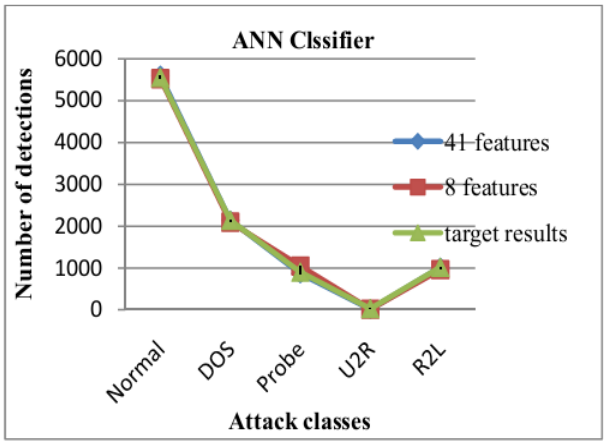


Fig. 4 Comparison of number of detections before and after feature reduction. (Hidden Layers=25 and Max Steps for Training taken =700) [8]

Njla and Hana [17] make a good study by using 5, 10, 21, 41 features. Their results show that high detection rate can be achieved with 5 features fig. (5). It is clear that depend on less number of features give high detection rate and take little processing time fig. (6).

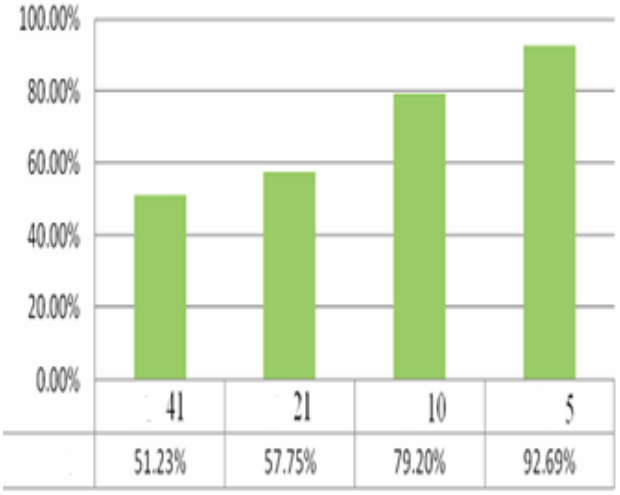


Fig. 5 Detection rate for 5, 10, 21 and 41 features [17]

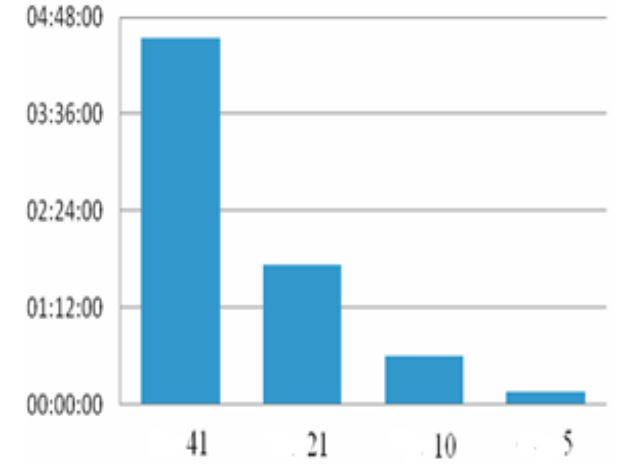


Fig. 6 Training time for 5, 10, 21 and 41 features [17]

For importance, it must be mentioned that evaluate high detection rate by depending on a few features drive to increase the false alarm because make the decision based on a part of the features that recognize the packet attack or not. This approach make the classification confused that classify clean packets as attacks. Therefore, there are a tradeoff between the detection rate with time taken and the false alarm [18].

III. THE PFE DESIGN

According to the previous studies that mentioned up, 12 features are chosen to be the target for the PFE. The chosen features include the most effect features on the detection of DOS, U2R and R2L attacks that suggested in the table1 [12], [14]. Table2 shows the selected features and the order number of each feature in the original NSL-KDD dataset. The selection was based on the dependency rate that the attacks detection depends on which feature. The selected features related with DOS, R2L and U2R attacks only, the PROBE attack not included. To design the Extractor, C#.Net is used. C#.Net is a strong programming language designed especially for the network applications. It is high flexibility, very simple and the most powerful Microsoft Visual Studio languages.

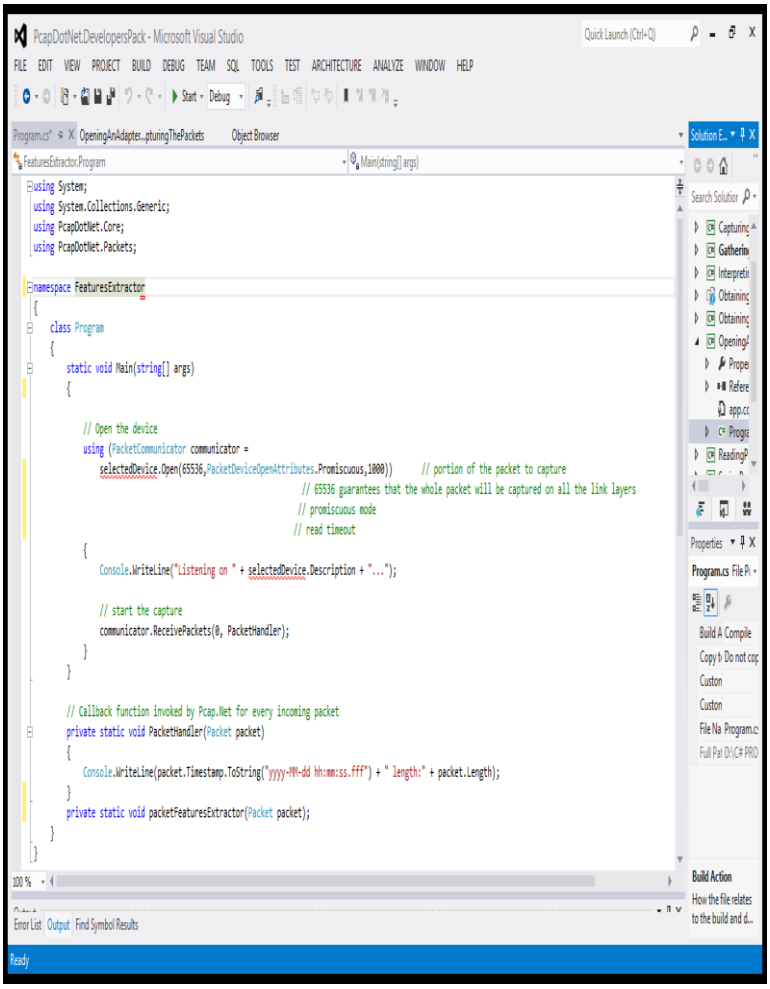


Fig. 7 Program piece code for the features extractor, M-soft VS2012-C#.Net

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

No.	Feature Name	Feature's order	Dependency Ratio [12]	Attack type	Attack class
1	source bytes	5	0.9708	Back	DOS
2	land	7	0.9999	Land	
	source bytes	5	0.9328	Neptune	
3	wrong fragment	8	0.9853	Pod	
	source bytes	5	0.7731	Smurf	
	wrong fragment	8	0.9913	teardrop	
4	count	23	0.6183	Smurf	
5	Duration	1	0.5682	guess_passwd	R2L
6	failed login	11	0.9622	guess_passwd	
7	service	3	0.9980	Imap	
8	destination bytes	6	0.9976	Phf	
	count	23	0.7898	multihop	
	destination bytes	6	0.7500	warezmaster	
	service	3	0.6658	ware client	
9	dst host srv serror rate	39	0.9997	Spy	
	service	3	0.6965	buffer overflow	U2R
10	dst host same srcport rate	36	0.6279	Load module	
11	root shell	14	0.9994	Perl	
12	srv count	24	0.7269	rootkit	

Table 2 the selected features based on the dependency ratio for the attack types

Table 3 Comparison detection Results with other works [1]

Owner	Used Method	Dataset	Result PSC %			
			R2L	DoS	U2R	AVG
Vaitsek-hovich[22] 2009	RNN & MLP	KDD-99	85.59	94.2	86.54	88.77
Laheeb [23] 2013	SOM ANN	KDD-99	91.86	93.61	92.14	92.37
		NSL-KDD	75.37	79.58	71.6	75.49
Prasanta Gogoi and etal. [24] 2013	Hybrid	NSL-KDD	89.14	99.19	66.67	85
Proposed	IBM	NSL-KDD	93.8	96.3	94.6	94.9

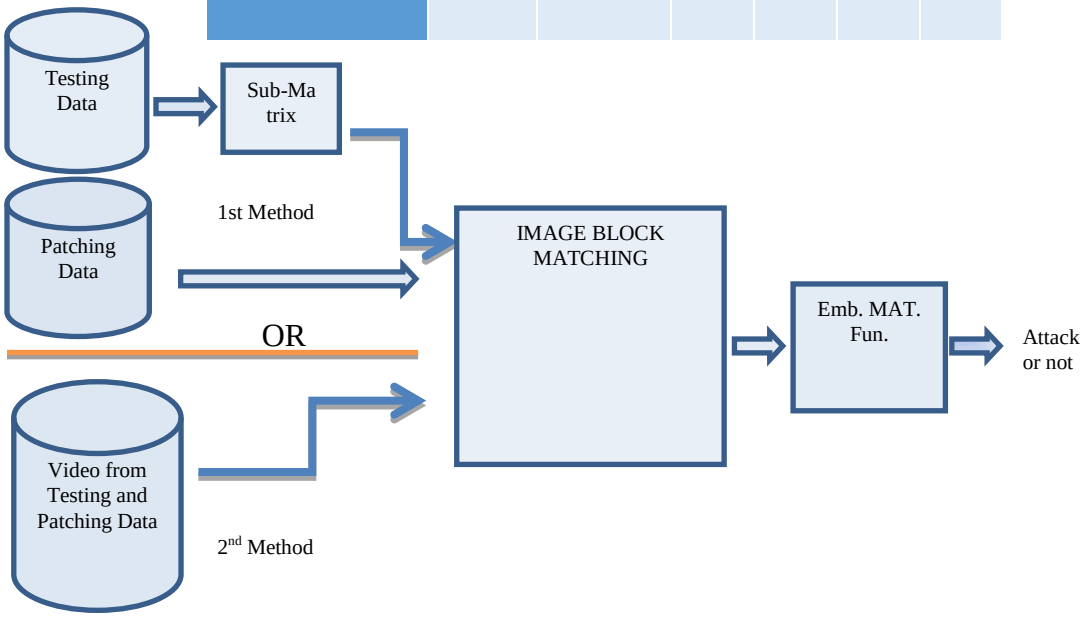


Fig. 8 IDS by using IBM [1]

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

Windows Packet Filter Kit – WinPKfilter, PacketX, Sharp Packet Capture – SharpPcap and Packet Capture.Net – PcapDotNet are the more popular libraries work on C#.Net. PcapDotNet is the best one of these libraries [19]. PcapDotNet and SharpPcap are used in the Extractor designing. First step is sniffing the packet and then temporary storing it then the extracting function is called to complete the job fig. (7). The PFE that designed here is a complementary part for a previous work [1] [20]. Image Block Matching is used as a System Core to design an off-line IDS fig.(8). The off-line IDS designed by using Simulink software and it evaluated a high detection rate about 94.9% based on 12 features from the NSL-KDD dataset that discussed above table2. Table3 shows a comparison of the evaluated results with the results of other researchers. The PFE designed to develop the off-line IDS [1] to be applicable on-line IDS for that, it extract the same 12 features that the off-line IDS designed on it. To use the designed PFE in the prebuild off-line IDS, it must be embedded in a Simulink block. C function block is used to pick up the C# Features Extracting project and it inserted in the system to replace the testing Dataset fig. (8). The features that was passed to the System Core from the dataset replaced with that extracted from the network line by the PFE.

States			Min. TET	Max. TET	Avg. TET
Run for the first time	Run in Simulink	Model-1	1.2 sec.	3.6 sec.	2.6 sec.
		Model-2	0.9 sec.	2.3 sec.	1.3 sec.
	Run in created OS.	Model-1	0.1 micro-sec.	5 micro-sec.	0.663 micro-sec.
		Model-2	0.01 micro-sec.	0.4 micro-sec.	0.311 micro-sec.
Run after the first time	Run in Simulink	Model-1	0.1 sec.	0.3 sec.	0.12 sec.
		Model-2	0.08 sec.	0.12 sec.	0.9 sec.
	Run in created OS.	Model-1	0.1 micro-sec.	5 micro-sec.	0.663 micro-sec.
		Model-2	0.01 micro-sec.	0.4 micro-sec.	0.311 micro-sec.

Table 4 the TET results of implementation the two models in the Simulink and RTOS [20]

IV. THE IMPLEMENTATION OF THE DESIGNED FEATURES EXTRACTOR

Network Systems need to be very speed in performance and not a bottleneck in the network line [21], for that, a Real Time Operation System (RTOS) had been created previously [20]. xPC Target Kernel was the base for the created RTOS. It able to load the models that built by the Simulink and run it with high performance. The RTOS tested by tow models UDP transmit/receive and DSP spectrum analyzer, fig. (9) Shows a snapshot of the UDP transmit model that applied on the RTOS.

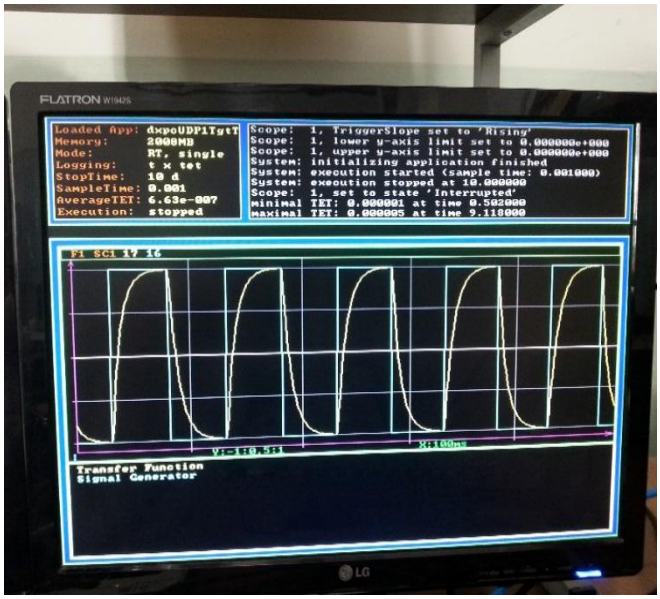


Fig. 9 The execution of UDP Model in the created RTOS [20]

The result shows a large Task Execution Time (TET) deference between running the models inside the Simulink environment and RTOS as shown in table 4. The PFE applied on the RTOS and tacked about (0.65 micro sec) average TET. The IDS, after insertion the PFE, is applied on the RTOS and achieved a very good throughput reached 99.97 Mbps in average. The throughput is limited by the Network Interface Card (NIC) properties. The RTOS has limited NIC vendors when it deal with TCP communication, it depend an Intel 8255x NIC and there was a 100Mbps NIC only in our experiment, so it is used.

V. RESULT AND CONCLUSIONS

The packet features are the base article of the network systems job, which include filtering and analyses. In this paper, a Packet Features Extractor (PFE) is produced. PFE is the most important part in the on-line NSS and it is the driver that through it the off-line system become an on-line system. The work that produced in this paper is an extended and developing for a previous works [1, 2, 20]. NSL-KDD features are depended in the designing of the PFE that extract 12 features of them. C#.Net is used to design the PFE and then it picketed inside a C Simulink block to be tested in off-line IDS that designed in a previous work and it was depend on the same 12 features [1]. The PFE drove the new on-line system and it gave the same detection rate that the old system (off-line IDS) was giving in Table3. The new online system loaded in a Real Time Operation System (RTOS) that created in a previous work [20]. The implementation of the system give a good result with a high performance and throughput about 100Mbps. It thought that it can be over that if a gigabit NIC is used. The throughput is limited by the NIC card but not by the system performance. Figure (10) shows that the average of the throughput saturated at 99.7 Mbps where the convergence values for the three deferent models

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

that simpler than the PFE. The values reach the end of the NIC limit which about 100Mbps. The system task (PFE over TCP communication) take about (0.65 micro sec) as average Task Execution Time (TET).

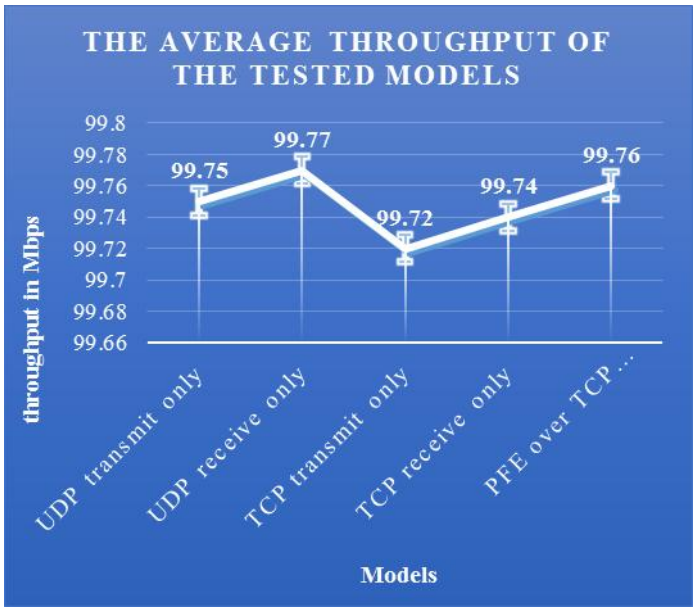


Fig. 10 comparison of the throughput results

REFERENCES

[1] Abdullah A. Mohamed, "Design Intrusion Detection System Based On Image Block Matching", International Journal of Computer and Communication Engineering,IACSIT Press, Vol. 2, No. 5, September 2013.

[2] Abdullah A. Mohamed, Dia M. Ali; "Challenges Of Designing And Implementing Intrusion Detection System", The International Conference on Computer Related Knowledge (ICCRK'2013), Sousse-Tunisia, IEEE Xplore Digital Library, in press Jun-2014.

[3] Ahmed P. and etal., "An intrusion detection and prevention system in cloud computing: A systematic review", Journal of Network and Computer Applications, Elsevier, Vol. 36(1), PP. 25–41, January 2013.

[4] Azmi, Reza; Pishgoo, Boshra., "SHADuDT Secure hypervisor-based anomaly detection using danger theory", Computers and Security, Elsevier, Vol. 39, Nov. 2013.

[5] Chirag Modi and etal., "A survey of intrusion detection techniques in Cloud", Journal of Network and Computer Applications, Elsevier, Vol. 36(1), PP. 42–57, January 2013.

[6] J. McHugh, "Testing Intrusion Detection Systems A Critique Of The 1998 And 1999 DARPA Intrusion Detection" ACM Transactions on Information and System Security, (TISSEC) 3 (4) (2000) 262–294.

[7] Hung-Jen Liao and etal., "Intrusion detection system: A comprehensive review", Journal of Network and Computer Applications, Elsevier, Vol. 36(1), PP. 16–24, January 2013.

[8] S. Lakhina, Sini J., B Verma, "Feature Reduction using Principal Component Analysis for Effective Anomaly–Based Intrusion Detection on NSL-KDD",International Journal of Engineering Science and Technology,Vol. 2(6) , 1790-1799, 2010.

[9] Oatley and etal., "SMART software for decision makers KDD experience", Knowledge-Based System, Elsevier, Vol. 15(5), PP. 323–333, Jul. 2002.

[10] Hee C., B. Jo, S. Choi, T. Park, "Feature Selection for Intrusion Detection using NSL-KDD", Recent Advances in Computer Science, ISBN: 978-960-474-354-4, 184-187, 2013.

[11] Shaheen A. , "A Comparative Analysis Of Intelligent Techniques For Detecting Anomalous Internet Traffic", MSc. Thesis, King Fahd University. 2010.

[12] A. Olusola., Adeola S. and D. Abosede, "Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features", Proceedings of the World Congress on Engineering and Computer Science, San Francisco, USA, October 20-22, 2010.

[13] Rupali D.,Shilpa L., "Performance Comparison of Features Reduction Techniques for Intrusion Detection System", International Journal of Computer Sience and Techenology", Vol. 3(1), 2012.

[14] Hafiz Muhammad Imran and etal., "Intrusions Detection based on Optimum Features Subset and Efficient Dataset Selection", International Journal of Engineering and Innovative Technology, Vol. 2(6), 2012.

[15] Taisir E., Mohammad K. and Aws K., "On The Kdd'99 Dataset: Statistical Analysis For Feature Selection", Journal of Data Mining and Knowledge Discovery, pp 88-90,Vol. 3(3), 2012.

[16] Ricardo A. and Rajesh S., "Feature Ranking and Support Vector Machines Classification Analysis of the NSL-KDD Intrusion Detection Corpus", Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, 2013

[17] Hana M. and Najla B., "Analysis Of Basic Compounds In Network Intrusion Detection System Based On NSL-KDD Dataset", Journal of AlRafidain for Computer Science and Mathematics", vol. 10(1), 2013.

[18] Georgios P.; Sokratis K., "Enhancing IDS performance through comprehensive alert post-processing", Computers and Security, Elsevier, Vol. 37, Sep. 2013

[19] Shamil Qays, "Design And Implementation Of a Security Software System Based On The Iris And Covert Channels", Msc Thesis, Mosul University, 2011.

[20] Abdullah A. Mohamed, Dia M. Ali, "Creating Real-Time Operation System Based on xPC Target Kernel", International Journal of Recent Technology and Engineering, Volume-2, Issue-4, September 2013.

[21] Po-Ching Lin and etal., "Re-examining the performance bottleneck in a NIDS with detailed profiling", Journal of Network and Computer Applications, Elsevier, Vol. 36(2), PP. 768–780, March 2013.

[22] Vaitsekhovich L. "Intrusion Detection in TCP/IP Networks Using Immune Systems Paradigm and Neural Network Detectors". Brest State Technical University, XI International PhD Workshop, OWD 2009. http://www.cs.ucc.ie/misl/publications/files/idssteinebach.pdf.

[23] Laheeb M., "A Comparison Study For Intrusion Database (Kdd-99, Nsl-Kdd) Based On Self Organization Map (Som) Artificial Neural Network ", School of Engineering, Taylor's

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

University, Journal of Engineering Science and Technology, Vol. 8(1), pp 107-119, 2013.

[24] Prasanta Gogoi and etal. , "MLH-IDS: A Multi-Level Hybrid Intrusion Detection Method", The Computer Journal Advance Access, Published by Oxford University Press on behalf of The British Computer Society, 2013.

AUTHOR'S PROFILE

Abdullah A. Mohamed Alsabbaui: born in Mosul-Iraq in 1988. He received his engineering degree (MSC)in Communication Science from the University of Mosul in 2011. He is a student in master study of network security. His published papers are "Designing of Intrusion Detection System Based on Image Block Matching", "Challenges of Designing and Implementing Intrusion Detection System" and "Creating Real-Time operation System Based on xPC Target Kernel".

Dia M. Ali: received his BSC in Elec. Comm. Eng. From Mosul university in 1992 (Mosul/ Iraq). In 1998, he gets MSC in Elec. Comm. Eng. Mosul University (Mosul / Iraq) and 2007-he received a PhD in Comm. Eng. (Network) from Mosul University (Mosul / Iraq). From 1993 to 2001, he worked at R&D Center Mosul / Iraq. Since 2007 join university of Mosul collage of Engineering as lecture. He interesting in Network Modeling and simulation especially in (OPNETMODELER), Network Security, Mobile Network planning, Antenna modeling and System.