ISBN 978-93-84468-11-8
Proceedings of 2014 International Conference on Artificial Intelligence & Manufacturing Engineering (ICAIME 2014)
Dubai, December 25-26, 2014, pp. 169-177

An enhanced Hybrid Anomaly-based Detection Approach

Tamer F. Ghanem[1], Wail S. Elkilani[2], Hatem S. Ahmed[3], and Mohiy M. Hadhoud[4]
[1, 4] Information Technology Dept., [2]Computer Systems Dept., [3] Information Systems Dept.
[1, 2, 3, 4] Faculty of Computers and Information
[1]tamerfg@hotmail.com, [2]wail.elkilani@gmail.com, [3]hatem6803@yahoo.com, [4]mmhadhoud@yahoo.com

Abstract: *During the last decade, Intrusion Detection Systems (IDSs) have played an important role in defending critical computer systems and networks from cyber-attacks. Anomaly detection techniques have received a particularly great amount of attention because they offer intrinsic ability to detect unknown attacks. In this paper, we propose an enhanced hybrid anomaly detection approach based on negative selection algorithm and metaheuristics. The enhancements include tuning some of its parameters value automatically without predefining them. NSL-KDD dataset; which is a modified version of the widely used KDDCUP99 dataset; is used for performance evaluation. KDDCUP99 dataset is criticized by its inability to reflected recent network traffic behaviour. So, a real time experiment was performed to capture and construct a recent dataset to ensure the performance of the proposed enhancements. Performance evaluation shows that the proposed approach outperforms other competitors of machine learning algorithms on both datasets.*

Keywords: *Intrusion detection; Anomaly detection; Negative selection algorithm; Intrusion detection datasets.*

1. Introduction

In recent years, numerous security issues are raised duo to the wide usage of computer networks which affect services availability for legitimate users. As a result, intrusion detection systems (IDSs) are expected to play a significant role in defending these intrusive activities [1].

Intrusion detection approaches are classified into either signature-based or anomaly-based. Signature-based approaches use signatures for matching and detecting attacks. They are known with their high accuracy for known attacks but they are incapable of detecting new attacks. On contrary, anomaly-based detectors try to build models for normal behavior in order to detect any deviation from the built model. They are able to detect previously unseen attacks but with less accuracy and higher false positive rate (FPR, events incorrectly classified as attacks) [2]. Techniques like statistical methods, data mining, machine learning algorithms, and evolutionary algorithms could be used for building detection models based on the use of training intrusion detection datasets.

KDDCUP99 is an intrusion detection dataset for IDS evaluation which is the most widely used because of its connections are well labelled and large number of attacks is included. Despite of this, it is considered outdated dataset because it does not reflect perfectly the recent behavior of normal and malicious traffic activities [3].

In previous research [4], we proposed a hybrid approach to anomaly detection using some inspiration of negative selection algorithm. Its detector generation (model building) is based on using k-means clustering along with a multi-start metaheuristic method and a genetic algorithm. Its detectors are shaped as hyper-spheres with radii thresholds to cover normal volumes. In another previous research [5], we proposed a fast clustering algorithm for large scale high dimensional dataset called Density Partitioning and Merging (DPM). Results shows its extreme fastness compared to other clustering algorithms. Its fastness comes due to its ability to simplify clustering operations with the help of three insensitive parameters.

In this work, DPM is integrated into the previously proposed intrusion detection approach and an enhancements are added to the detector generation process for the sake of lessen the overall processing overhead and to automate value selection of two parameters of the proposed intrusion detection approach. In addition, a recent network traffic is captured and analyzed to build a new intrusion dataset that reflects the current behavior of normal and malicious network activities. Performance evaluation of the enhanced intrusion detection approach on both KDDCUP99 and the built real dataset shows its superiority compared to other machine learning algorithms and other proposed detection approaches in literature.

The rest of this paper is organized as follows: Section 2, introduces briefly the previously proposed anomaly detection approach and DPM clustering algorithm. Section 3 presents some literature review on anomaly detection approaches. Section 4 discusses the enhanced intrusion detection approach and real dataset construction. Experimental results along with a comparison against other machine learning algorithms and other proposed approaches in literature are presented in section 5 followed by some conclusions in section 6.

2. Previously proposed research

1.1. Previously proposed intrusion detection approach.

Detector generation process in the previously proposed intrusion detection approach [4] goes through a number of stages in order to generate anomaly detectors with high performance as stated below.

Preprocessing: Training dataset (*DS*) is normalized to be ready for later processing.

Clustering and Training Dataset Selection: In this step, *DS* is clustered into k clusters using k-means in order to select a small sample training dataset (*TR*) with size of *sz* samples with a good representation of *DS*. This is done by distributing the selection of (*TR*) samples over the labeled (*DS*) classes and clusters in each class.

Detector Generation using Multi-Start Algorithm: Multi-start searching algorithm is used for generating detectors (hyper-spheres) which is used later to detect anomalies. Hyper-spheres detectors are used and defined by its center and radius. A number of well selected initial start points (*isn*) is used for enhancing solutions diversity.

Detector Radius Optimization using Genetic algorithm: This stage is used to optimize only detectors radius to cover the maximum possible number of only normal samples using genetic algorithm.

Detectors Reduction: Redundant detectors are removed to improve the speed of later online anomaly detection.

Repetitive Evaluation and Improvements: Detector generation and reduction stages are repeated until a satisfied condition exists. Two iterations are enough to obtain stable results.

1.2. Previously proposed DPM clustering algorithm.

DPM is a fast clustering algorithm for large scale high dimension datasets with three steps as stated below [5].

Dimension based data partitioning: Data is partitioned into small dense partitions based on the successive processing of each dimension histogram.

Noise filtering: Noise detection is accomplished by estimating dimensional density of all partitions generated in the first stage. Then partitions that have low dimensional densities are considered noise and removed.

Partitions merging and clusters construction: Boundary samples of each generated partition are used to make a decision about merging with nearest partitions in order to form clusters.

3. Related work

Statistics-based approaches are one several categories used in anomaly detection. They uses means like predefined threshold and probabilities to identify abnormal activities. Another category is Rule-based approaches which use If–Then–Else rules for building detection models. State-based approaches use finite state machine to describe network behavior and detecting attacks [6].

In recent years, negative selection algorithms (NSAs) are used to identify malicious activities [7] based on its detector generation scheme. Genetic and particle swarm algorithms, are the widely used algorithms in generating NSA detectors. In [8], genetic-based NSA is used for generating Hyper-sphere detectors based on a fitness function that maximizes the coverage of no-self (abnormal) space. Another genetic based NSA with deterministic crowding niching technique is introduced in [9] for improving hyper-sphere detectors generation. Deterministic crowding niching is used with genetic for the purpose of improving the diversification and generating high quality solutions. Another work is presented in [10] and [11], which tests the usage of hyper-rectangular and hyper-ellipsoid detectors in anomaly detection using evolutionary algorithm. Another approach is proposed in [12]. This approach begins by feature selections based on rough set and then a modified version of particle swarm algorithm is used for detectors generation. Another work for detecting anomalies hidden in the self-regions using boundary detectors is proposed in [13]. This approach uses evolutionary search algorithms for detectors generation.

In this research, an enhanced intrusion detection approach is introduced and tested using the widely used intrusion detection dataset KDD CUP 99. In addition, a real traffic dataset is constructed and labeled in order to reflect current network traffic behavior. This dataset is used for evaluating the proposed approach enhancements to ensure its high performance in recent networks. Results shows the superiority of the proposed enhanced approach compared to other machine learning algorithms and other literature proposed approaches.

4. Methodology

1.3. The Enhanced intrusion detection approach

Although k-means clustering algorithm is known with its fastness, it is not suitable for clustering large scale datasets in terms of time consumption and resources needed to accomplish clustering process. It is one of the reasons that affects the overall processing time overhead of the proposed intrusion detection approach. Here is where DPM clustering is used to enhance the processing overhead caused by the earlier usage of k-means. As shown before, DPM is characterized by its fastness and its capability of clustering huge amounts of data effectively.

Furthermore, detector generation stage using multi-start algorithm are modified to enhance the overall processing time. Multi-start method is one of metaheuristic methods which is used to obtain a number of high quality solutions in large solution space (e.g. intrusion detection space). The basic concept of multi-start method is simple: start searching for solutions using local solvers initiated from multiple well-selected starting points, in hopes of locating solutions of better quality (which have smaller objective function values by definition). Clustering algorithm is used to select a number of good initial starting points which is distributed over the generated clusters.

First of all, multi-start is modified to minimize the searching space for each initial start point. The previously proposed approach uses the upper and lower bound of the whole solution space to be the boundary of solution searching process for each initial start point. This may lead to a bigger chance to obtain some new solutions that may be near from the each other and so, they will be removed later in detectors reduction stage. Moreover, duo to the largeness of the searching space, this may lead to obtain some solution with poor quality. To enhance the detector generation process, each solution searching process will be bounded by lower and upper values of the cluster that its initial start point belongs to. This is done as the following:

$clusters\ (C) = \{C1, C2, C3, \ldots, Ck\},$

$data\ source\ (DS) = \{x_{ij} | i = 1,2,3, \ldots, m\ and\ j = 1,2,3, \ldots, n\},$

$u_j^c = max(x_j^c)$

$l_j^c = min(x_j^c)$

$upper\ bound\ (UB^c) = (u_1^c, u_2^c, u_3^c, \ldots, u_n^c, rrl),$

$lower\ bound\ (LB^c) = (l_1^c, l_2^c, l_3^c, \ldots, l_n^c, 0),$

Where *k* is number of clusters, *DS* is the data source of m samples and n columns, *rrl* is the upper limit of detector radius, cluster number $(c) = 1,2,3,\ldots, k,$ and x_j^c is the j*th* column of all samples that belong to cluster number *c*.

Second, multi-start objective function is modified to minimize the number of samples to work on for obtaining each solution (detector). This is because all samples in solution space are processed whenever objective function is computed in the previously proposed approach which implies more processing overhead. Good solutions (detectors) are defined as those solutions that cover the maximum number of normal samples, minimum number of abnormal samples and with lower intersection with previously generated solutions. Objective function is modified to count the covered normal and abnormal samples within only the cluster where the initial start point belongs to. As a result, the number of processed samples at each execution of the objective function is minimized and the overall processing overhead is enhanced. The objective function is modified to be as the following:

$$f(s_i^c) = \begin{cases} N_{abnormal}^c(s_i^c) - N_{normal}^c(s_i^c) & , \quad itr = 1 \\ N_{abnormal}^c(s_i^c) - N_{normal}^c(s_i^c) + \\ old\_intersect^c(s_i^c) & , \quad itr > 1 \end{cases} \quad (1)$$

where solutions s_i^c is i*th* solution generated from cluster number *c*, and each solution is in the form of $s_i^c = (u_{i1}, u_{i2}, u_{i3}, \dots, u_{in}, r_i)$, where hyper-sphere center is at $S_{center} = (u_{i1}, u_{i2}, u_{i3}, \dots, u_{in})$, *n* is the number of dimensions, hyper sphere radius is r_i, $N_{normal}^c(s_i^c)$ and $N_{abnormal}^c(s_i^c)$ is the number of normal and abnormal samples covered by solution s_i^c respectively.

Moreover, at higher iterations *(itr > 1)*, the intersection between the newly generated solution (s_i^c) and old solutions $(old\_intersect^c(s_i^c))$ is computed using only the old solutions generated from the same cluster rather than using all old solutions. This enhances processing overhead for doing such computation.

Finally, as mentioned earlier, four parameters affect the performance of the previously proposed intrusion detection approach. These parameters are training dataset size (*sz)*, number of initial start points (*isn*), detector radius upper limit (*rrl*) and number of clusters (*k*). Two of these parameters, which are *k, isn*, are chosen to automate the selection of their values. Clusters number (*k*) value is set using DPM. Unlike k-means, DPM has the capability to detect clusters number automatically. Also, previous extensive study shows that higher number of initial start points (*isn=300*) and medium number of clusters (*k=200*) are preferable. As a result, number of selected initial start points is set to be as : *isn = (300/200) * k = 1.5 * k*.

1.4. Real dataset construction experiments

Although KDDCUP is the most widely used dataset for evaluating intrusion detection systems, it may not reflect perfectly the current network traffic behavior and intrusion patterns as they evolve. So, real network traffic is captured and analyzed to construct real intrusion detection dataset for the sake of performance evaluation of the enhanced proposed system.

Real traffic is captured from the network of Faculty of Computers and Information, Menofia University. Network structure is prepared as shown in Figure 1.This network includes wired and wireless connected users. In addition, server farm is built to enable email, file and web services to enrich the services in the network. A monitoring zone is added and configured for tapping traffic passes between internal devices and internet. Tcpdump tool is used for traffic capture. SNORT, BRO, and OSSIM IDSs software with their latest updates are used to analyze and detect attacks in the captured traffic. In addition, some scripts and tools are used to generate a number of denial of service (DOS), probing, and remote to local (R2L) attacks against internal users and services. Also real internal attacks duo to internal compromised devices and internet attacks against internal attacks are captured. This structure gives the ability to capture real normal and abnormal network activities.

The captured packets is then analyzed and grouped to form connections. Then, a script is built to extract 25 KDDCUP features for each connection. Finally, Attacks is detected using SNORT, BRO, and OSSIM software and matched with the formed connections for the sake of labelling each connection as normal or attack.

Connections statistics of the captured datasets is shown in Table 1. It is notices that some connections may include more than one attack. Also, attack connections occupy only 3.94% of total connections. Statistics of these attacks are stated in Table 2. These statistics shows that network Trojan attacks and privacy violation attacks are the major attacks that face legitimate network users. This means that new attacks with different behavior are added to network activities compared to KDDCUP99 dataset where DOS and probing attacks was the dominants.

In Figure 2, statistics of dataset protocols and services are stated. TCP and UDP are the main used protocols in the captured traffic. It is noticed that web, remote desktop, and video streaming are the main TCP services request by network users. Also, DNS and streaming are the most used UDP services in the captured dataset.

TABLE I: Connection statistics of the captured dataset.

		%
Total Connections	1063226	
Capture Duration	one week	
Number of packets	22353001	
Total Connections	1063226	100.00%
Normal connections	1021385	96.06%
Abnormal connections	41841	3.94%
Connections with one attack	38330	3.61%
Connections with multiple attacks	3511	0.33%
Total number of attacks	49579	

TABLE II: Attacks statistics of the captured dataset

Attack category	Connections	%
Network Trojan	35557	71.72%
Privacy violation	13691	27.61%
Bad traffic	164	0.33%
DOS&probing	91	0.18%
Web attacks	59	0.12%
Privilege gain trials	9	0.02%
Misc Attack	4	0.01%
Executable code	3	0.01%
Information leak trial	1	0.00%
Total	49579	100.00%

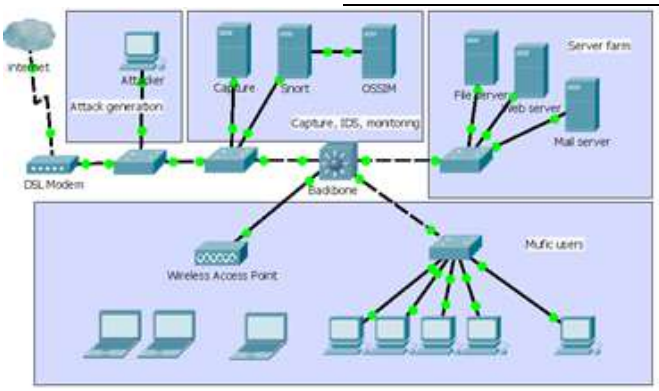

Fig. 1: Network structure used for building real traffic dataset.

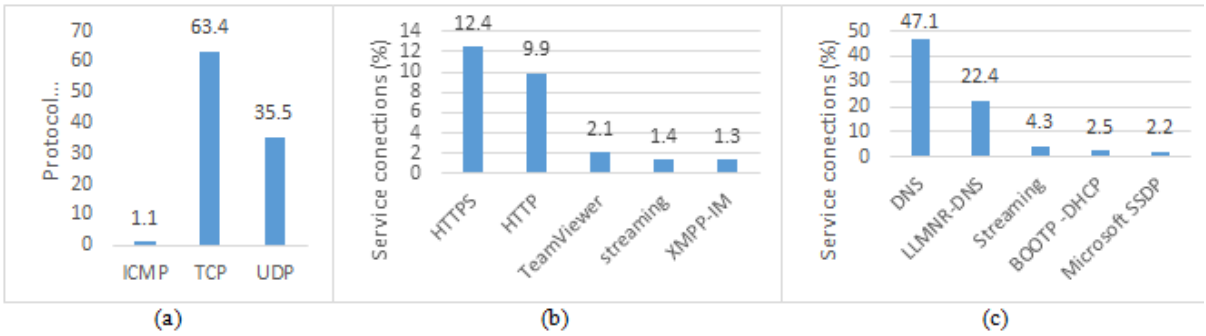

Fig. 2: Captured dataset statistics. a) Protocol connections statistics. b) Top 5 TCP services connections relative to total TCP connections. c) Top 5 UDP services connections relative to total UDP connections

5. Results and discussions

1.1. Experimental setup

In this experiment, NSL-KDD; which is a modified version of KDDCUP99 [3]; and real built dataset is used for evaluating the enhanced intrusion detection approach. NSL-KDD has a number of more than one million of network connections with 41 features plus connection label. Each connection label belongs to one of five main classes (Normal, DOS, Probe, R2L, and U2R). NSL-KDD dataset includes training dataset *DS* with 23 attack types and test dataset *TS* with additional 14 new attacks not exist on training dataset. Our experiments ran on a system with 3.0 GHz Intel® Core™ i5 processor, 4GB RAM and Windows 7 as an operating system. As proposed in [5], DPM is controlled by three insensitive parameters. This means that DPM results does not vary over a range of values for each of these parameters. As a result, DPM parameters, which are dimension divisions, dimensional dense threshold and histogram peak detection sensitivity, is set to values of 100, 0.1 and 0.1 respectively as used in that research. Also hyper-sphere radius upper limit is set to 2 as recommend in [4].

Performance evaluation is done using three metrics which are processing time, classification accuracy and false positive rate (FBR). The last two metrics are calculated as follows:

$$classification\ accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2)$$

$$false\ positive\ rate = \frac{FP}{TN+FP} \quad (3)$$

Where true positive (*TP*) is normal samples correctly classified as normal, false positive (*FP*) is Normal samples incorrectly classified as abnormal, true negative (*TN*) is abnormal samples correctly classified as abnormal and false negative (*FN*) is abnormal samples incorrectly classified as normal.

1.2. Results

Processing time overhead comparison between the old hybrid anomaly detection (HAD) and the enhanced anomaly detection approach (E-HAD) is introduced in Table 3. In E-HAD, DPM clusters dataset into 166 clusters which is near the recommended number of clusters in [4]. The detected number of clusters by DPM is used as input to k-means clustering algorithm in old HAD to make fair comparison. Results shows that E-HAD has lower processing in both clustering time and detector generation using multi-start compared to the old HAD approach using different sizes of reduced training samples of NSL-KDD dataset.

TABLE III: Processing overhead comparison between enhanced and old approach using NSL-KDD dataset.

	HAD					E-HAD				
samples size	multi-start	radius optimizing	rule reduction	k-means	total (seconds)	multi-start	radius optimizing	rule reduction	DPM	total (seconds)
10000	35.4	39.0	1.5	71.3	147.2	28.7	48.7	2.4	30.7	110.5
20000	132.3	126.9	7.0	71.3	337.5	67.0	174.9	13.5	30.7	286.1
40000	284.6	251.2	18.0	71.3	625.1	67.2	285.2	27.9	30.7	411.0

In Figure 3, performance comparison between E-HAD, HAD and six of other machine learning algorithms used for intrusion detection is presented. These algorithms are Bayes Network (BN), Bayesian Logistic Regression (BLR), Naive Bayes (NB), Multilayer Feedback Neural Network (FBNN), Radial Basis Function Network (RBFN), and Decision Trees (J48). Results shows that E-HAD has higher accuracy than HAD with nearly the same FBR at different NSL-KDD training dataset sizes. But E-HAD has lower processing overhead compared to HAD approach. In addition, E-HAD has the highest accuracy as well as the lowest FBR along with moderate processing time overhead compared to other machine learning algorithm.

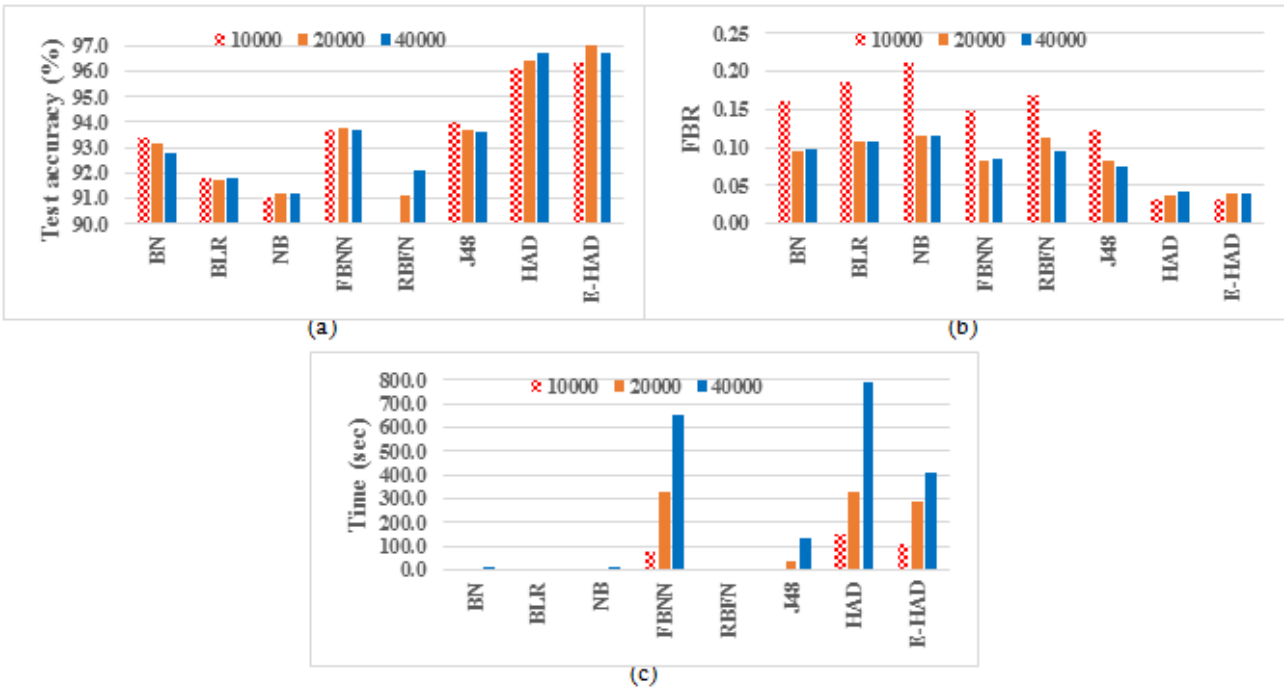

Fig. 3: Performance comparison of the enhanced model using NSL-KDD dataset.

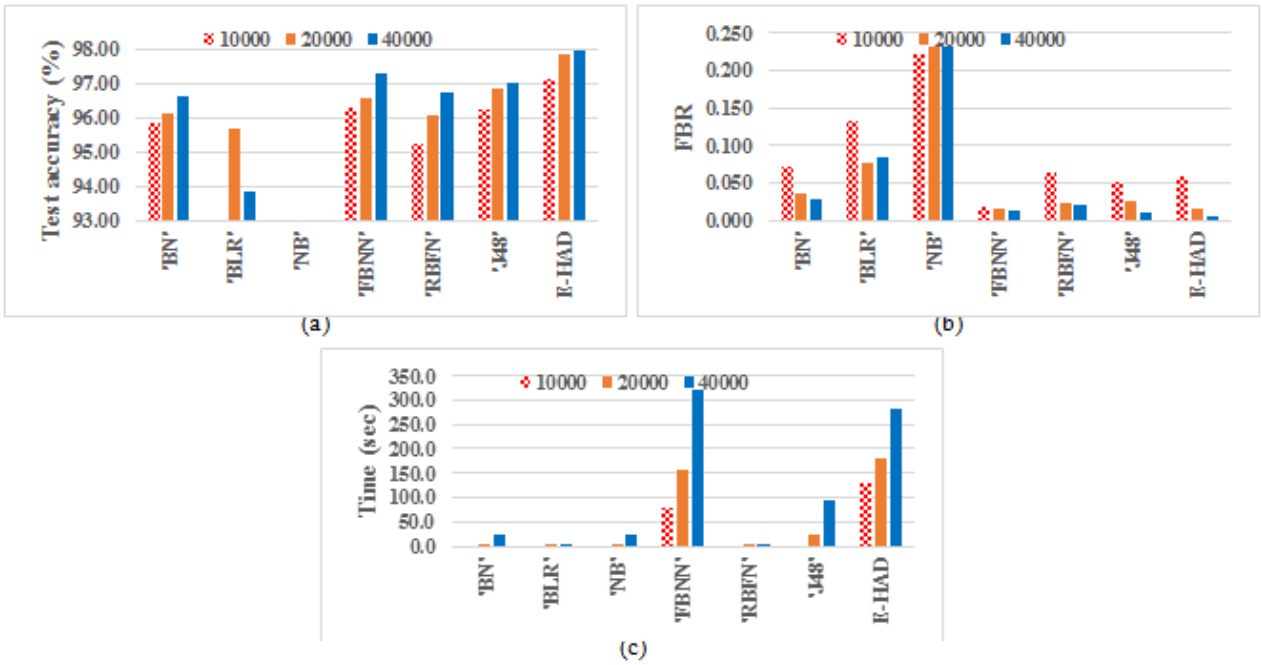

Fig. 4: Performance of the enhanced model on the real captured dataset.

Performance evaluation of E-HAD is tested using the built real dataset as introduced in Figure 4. Results shows that E-HAD has the highest accuracy as well as the lowest FBR at training datasets of size 20000 and 40000 samples compared to other algorithms. Although E-HAD has more processing overhead at offline training, a positive effect is expected at online detection due to generating detectors of high detection accuracy and low false positive rate.

TABLE IV: Comparison with other research work based on KDDCUP99 dataset.

Ref.	Approach	Train size	Test size	KDDCUP test set	Accuracy %	TPR %	FPR
[14]	FtNSA (GA)	7000	34000	no	-	96.2	0.024
	NSA (GA)	7000	34000	no	-	97.5	0.080
	SDC (GA)	7000	34000	no	-	96.6	0.070
	E-HAD	**7000**	**all test set**	**yes**	**-**	**96.0**	**0.028**
[12]	SSO-WLS (rough sets, PSO)	4000	4000	no	93.3	-	-
	SSO (PSO)	4000	4000	no	89.6	-	-
	PSO (PSO)	4000	4000	no	88.5	-	-
	Naive Bayes	4000	4000	no	92.1	-	-
	SVM	4000	4000	no	86.8	-	-
	E-HAD	**4000**	**all test set**	**yes**	**96.0**		**0.025**
[9]	NSA (GA)	13449	22545	yes	81.8	-	-
	NB Tree	13449	22545	yes	82.0	-	-
	Multi-layer neural network	13449	22545	yes	77.4	-	-
	SVM	13449	22545	yes	68.5	-	-
	E-HAD	**13449**	**all test set**	**yes**	**96.3**	**-**	**0.034**
[13]	NSA (GA)	48639	445112	no	-	93.0	0.080
	E-HAD	**48639**	**all test set**	**yes**	**-**	**96.8**	**0.044**
[15]	Model(Offline)(k-means, SOM)	20000	20000	no	-	89.7	0.243
	Model(online)(k-means, SOM)	20000	20000	no	-	96.6	0.130
	E-HAD	20000	all test set	yes	-	96.4	0.041
[16]	EFuNN(Evolving Fuzzy Neural)	38910	58947	no	-	85.0	0.009
	FART(Fuzzy)	38910	58947	no	-	94.0	0.030
	SVM	38910	58947	no	-	90.7	0.124
	E-HAD	**38910**	**all test set**	**yes**	**-**	**96.8**	**0.042**

A comparison is held between E-HAD and six of other proposed research work in the field of intrusion detection systems using KDDCUP dataset as shown in Table 4. Different research approaches are used in these papers which are inspired from different machine learning and evolutionary techniques. E-HAD is evaluated with similar training datasets according to each of these research papers. Some of these work is evaluated using only accuracy or using accuracy and FBR while others use true positive rate (TPR) and FPR as metrics for such

evaluations. Whatever the metrics used, E-Had is evaluated with the same metrics. All of these research work, except research in [9], is tested with partial datasets selected from the training dataset of KDDCUP99 and not from the test dataset of KDDCUP99. KDDCUP99 has extra 14 attack not existed in KDDCUP99 training dataset. On the other hand, E-HAD is tested with all samples of KDDCUP99 test dataset. Results shows that E-HAD performance keeps balancing between higher accuracy or TPR and lower FPR compared to other approaches which ensure its superiority compared to other research work.

6. Conclusion

In this research, An Enhanced anomaly detection approach is introduced. The enhancements include the integration between the previously proposed hybrid anomaly detection approaches with the previously proposed DPM clustering algorithm. In addition, Detector generation using multi-start metaheuristic is modified for the sake of minimizing the search space for each solution and minimizing the number of processed samples for each objective function computation. Moreover two control parameters, which are clusters number and initial start points number, are chosen to be select automatically.

Furthermore, a new recent intrusion detection dataset is constructed using real network traffic. A suitable network structure is implemented. Captured packets is then transformed to connections with 25 features for each of them. Then connections is labeled as normal or attack. This dataset reflects current traffic behavior which is missed by the old widely used intrusion detection dataset KDDCUP99.

In general, performance Evaluation or real and KDDCUP99 dataset shows that our enhanced approach has higher accuracy with lower FPR compared to other machine learning algorithms with more processing overhead in some cases. Despite of this processing overhead at offline training, a positive effect is expected at online detection due to the high detection accuracy and low false positive rate of the generated detectors.

In future research, more enhancements is needed to improve the processing overhead especially in detectors radius optimization stage. Also, more research is needed in order to make the enhanced anomaly detection approach is fully automated which increases it applicability in real world.

References

[1] Liao HJ, Lin CHR, Lin YC, Tung KY. Intrusion detection system: A comprehensive review. J Netw Comput Appl. 2013; 36(1): p. 16-24.
http://dx.doi.org/10.1016/j.jnca.2012.09.004

[2] García-Teodoro P, Díaz-Verdejo J, Maciá-Fernández G, Vázquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Comput Secur. 2009; 28(1–2): p. 18-28.
http://dx.doi.org/10.1016/j.cose.2008.08.003

[3] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In CISDA 2009 Proceedings of IEEE Symposium on Computational Intelligence for Security and Defense Applications; 2009; Ottawa, Canada: IEEE Press. p. 1-6.

[4] Ghanem TF, Elkilani WS, Abdul-kader HM. A hybrid approach for efficient anomaly detection using metaheuristic methods. J Adv Res. 2014.

[5] Ghanem TF, Elkilani WS, Abdul-kader HM. DPM: fast and scalable clustering algorithm for large scale high dimensional datasets. In 10th International Computer Engineering Conference ICENCO 2014; 2014; Faculty of Engineering, Cairo University, Giza, Egypt. p. in press.

[6] Liao HJ, Lin CHR, Lin YC, Tung KY. Intrusion detection system: A comprehensive review. J Netw Comput Appl. 2013; 36(1): p. 16-24.
http://dx.doi.org/10.1016/j.jnca.2012.09.004

[7] Dasgupta D, Yu S, Nino F. Recent Advances in Artificial Immune Systems: Models and Applications. Appl Soft Comput. 2011; 11(2): p. 1574-1587.
http://dx.doi.org/10.1016/j.asoc.2010.08.024

[8] Gao XZ, Ovaska SJ, Wang X. Genetic Algorithms-based Detector Generation in Negative Selection Algorithm. In SMCals/06 Proceedings of IEEE Mountain Workshop on Adaptive and Learning Systems; 2006; Utah, Logan, USA: IEEE. p. 133-137.

[9] Aziz ASA, Salama M, ella Hassanien A, El-Ola Hanafi S. Detectors generation using genetic algorithm for a negative selection inspired anomaly network intrusion detection system. In FedCSIS Proceedings of Federated Conference on Computer Science and Information Systems; 2012; Wroclaw: IEEE. p. 597-602.

[10] Ostaszewski M, Seredynski F, Bouvry P. Immune anomaly detection enhanced with evolutionary paradigms. In GECCO '06, Proceedings of the 8th annual conference on Genetic and evolutionary computation; 2006; New York, NY, USA: ACM. p. 119-126.

[11] Shapiro JM, Lamont GB, Peterson GL. An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection. In Beyer HG, editor. GECCO '05. Proceedings of the 2005 conference on Genetic and evolutionary computation; 2005; New York, NY, USA: ACM. p. 337-344.
http://dx.doi.org/10.1145/1068009.1068063

[12] Chung YY, Wahid N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). Appl Soft Comput. 2012; 12(9): p. 3014-3022.
http://dx.doi.org/10.1016/j.asoc.2012.04.020

[13] Wang D, Zhang F, Xi L. Evolving boundary detector for anomaly detection. Expert Syst Appl. 2011; 38(3): p. 2412-2420.
http://dx.doi.org/10.1016/j.asoc.2012.04.020

[14] Gong M, Zhang J, Ma J, Jiao L. An efficient negative selection algorithm with further training for anomaly detection. Knowl-Based Syst. 2012; 30(0): p. 185-191.
http://dx.doi.org/10.1016/j.knosys.2012.01.004

[15] Lee S, Kim G, Kim S. Self-adaptive and dynamic clustering for online anomaly detection. Expert Systems with Applications. 2011; 38(12): p. 14891-14898.
http://dx.doi.org/10.1016/j.eswa.2011.05.058

[16] Liao Y, Vemuri VR, Pasos A. Adaptive anomaly detection with evolving connectionist systems. Journal of Network and Computer Applications. 2007; 30(1): p. 60-80.
http://dx.doi.org/10.1016/j.jnca.2005.08.005