From Insect to Secure: Applying Nest-mate Recognition Approach of Honeybee to Improve Intrusion Detection System

Ghassan Ahmed Ali
College of Computer Science and Information System
Najran University, Najran, Kingdom of Saudi Arabia

Abstract

The need of detecting system that able to detect intrusion attempts from attacking the system is a very critical issue. Intrusion Detection System (IDS) aims to support the essential security issues via scrutinizing every entry and then provide a feedback for the user regarding the systems' situation. However, the difficulty that IDS faces to determine whether such action is either a malicious or a normal reduces the full benefit of IDS and make the area of IDS an attractive and open research field. In this paper, a nest-mate recognition system of honeybee which keeps the colony safe is investigated for improving the detection accuracy and performance of the IDS. The performance of the proposed IDS is evaluated using NSL-KDD data set. The experiments show that the performance of the proposed approach can detect novel intrusions and reduce false alarms.

1. Introduction

The importance of computer security is not new issue. However, the risks have been increasing and the countermeasure approaches are increasing too. Researches in computer security technologies remain obsession for many years of improvement and growth. However, it still needs a lot of hard work to settle the critical security problems. The number of attacks are increase in 2014 over 2013 and they expect the number to rise again in 2015 [1].

Many techniques are used to defense against attacks such as firewall, honeypot, and encryption etc. However, most of these systems are still susceptible to attacks and intrusions. The Intrusion detection system (IDS) is a framework that acts against intrusions and monitor the network state by detecting unauthorized usage, denial of services, and anomalous behavior. The key of IDS is to detect the intrusion. Then, after detected, it can be in some way to prevent the intrusion as claimed by [2].

The accuracy of detecting intrusion is directly depending on the accuracy of classification which is the first layer of IDS. Poor classification will result in the occurrence of intrusion and false alert [3]. A classification method is very important to obtain effective countermeasure against the intrusions.

The ability to recognize and detect intrusion is critical to the maintenance of the integrity of social insect colonies. Therefore, many researches take steps toward supporting computer security by understanding the methods underlying social insects' behavior system which face the same problems and see how there system works.

The crossover between the behavior of social insects and computer science can be declared as ‘‘. . . any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies . . .’’ by [4]. From studying how social insects perform tasks, we figure out such model to be used as a basis of development, either by enhancing the model or by adding non biological features to the model. The most important is the applicability of the model. The mimicry in all details is kind of exaggeration; to a certain extent, the similarity that it deduces to be useful should be the most concern.

The intelligent behaviors of honeybee have been developed to different models and methods which are applied for solving various types of problems. In the literature survey, some studies modeled the honeybee foraging or finding home to be used in optimization problem [2]. Other works have proposed models based on the marriage behavior of honeybee [5]. From these models there being extracted many features were being utilized by engineering and computer science [6].

In this paper, we focus on how the bees solve such security problems regarding the detection to crossover directly to IDS. The concern is on discrimination between innocuous and the intrusion by capturing the intrusion ones based on some techniques, which have been inspired from the nestmate-recognition system in honeybee.

2. Background

Current researches on IDS solutions emphasis particularly on employing two detection aspects: signature and anomaly detection. The idea behind studying of these two aspects is to define a new

approach that includes aspects of both signature and anomaly detection technique.

A key advantage of deploying signature-based IDS detectors is that they are a cost-effective compromise to develop and can efficiently detect instances of known attacks. Their effectiveness is strictly related to the extent to which IDSs are updated with the signatures of the latest attacks developed or by generalization. Moreover, the misuse detection systems offer an earlier knowledge of what network wants to identify and assuring a certain degree of security in a system [7].

Due to the effectiveness of signature-based detection, we assume that the first detector engine in our system is a signature-based detection engine as a core component which idealizes the Undesirable-Absent (UA) of nest-mate recognition of honeybee approach. We implement the misuse detector flagging by comparing the incoming data to the patterns, learned by computational artificial intelligence, thus creating a new dimension to misuse detector engine. In addition, we improve the accuracy of detection and manage the shortcoming of the false of alarm flagging.

The second technique is an anomaly-based detection engine which seeks for unusual or something rare. It works on the notion that abnormal activity or behavior is different from normal characterization then detects the deviations. The major advantage of anomaly-based detector is that the anomaly detector has the capability of detecting new types of intrusions without previous knowledge of attack details, and only requires normal data when building profiles [8].

The information that provided by anomaly-based detector can be used to define signatures for signature-based detection engines. This is what exactly processed in nest-mate recognition of honeybee approach; the anomaly-based detector filters out the abnormal records to be used further to train the signature-based detector in order to develop the detection coverage.

The main issue of anomaly-based detectors lies in its inability of managing the high number of false alarm [9]. According to [8], the improvement and researches currently are concerned on this problem in the field of improvement IDS.

It is clear that the more flexible of flowing data, the more control is available to improve the performance. The Nest-mate recognition approach combines misuse and anomaly detectors sequentially. It employs misuse at first followed by anomaly detector.

Misuse detector has high detection rate for known intrusions. It is responsible to detect pre-defined attacks based on their attack signatures. Therefore it recognizes the intrusions with low false alarm and filter out the attacks. The suspicious activities forward to anomaly detector. The task is to determine whether these suspicious activities belong to ‘normal’ or to an ‘abnormal’. Then, the abnormal activities forward to Filtering Decision (FD) to processes the data. The *FD* will later be used to train the misuse detector. Figure 1 displays the block diagram of combination misuse and anomaly detectors sequentially.

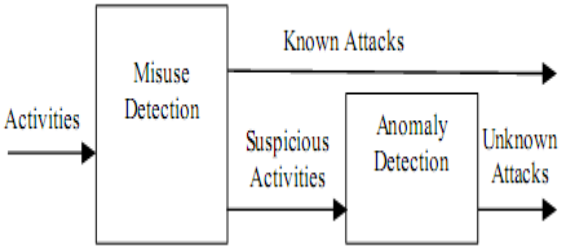

Figure 1. Combination of misuse and anomaly detectors

3. Nest-mate Recognition Approach

The components of nest-mate recognition approach are based on imitating a natural honeybee in detecting intrusions and using Neural Network (NN) trained by the improved Bees Algorithm (BA) for intrusion detection. The potential of trained NN in both anomaly and misuse detection can be demonstrated through the use of BA and the honeybee approach (Undesirable-Absent, Desirable-Present, Filtering Decision) as a basis for the inference engine detector.

The important tenets of the proposed detector for network intrusion detection lie in the new nest-mate recognition approach which imitating the natural honeybee in detecting the non-nest mate and direct it to a desirable behavior for network security, specifically, IDS. The idea here is to train the NN under the nest-mate recognition approach to detect the intrusions. The NN is trained by using the Bees Algorithm.

After the training phase, the NN is able to make the distinction between both normal and anomalous and of different attack classes. The general design and the architecture of the proposed inference engine draws inspiration and combines advantages of both modular hybrid detection method (anomaly and misuse) and artificial intelligence.

The principle interest of this work is to evaluate the performance of the proposed IDS by using KDD 99 dataset as a benchmark dataset which used by IDS researchers. In addition to that, the KDD 99 is used as the main intrusion detection dataset for training. The goal is to train the system with different types of attacks data and model different types of attack signatures.

The hierarchical hybrid strategy of the proposed approach is illustrated in Figure 2. The latter Figure demonstrates a streamlined, centralized intrusion detection design that consists of both the misuse detection method and the anomaly detection method. The workloads are distributed to the different detectors to monitor the intrusion detection process.

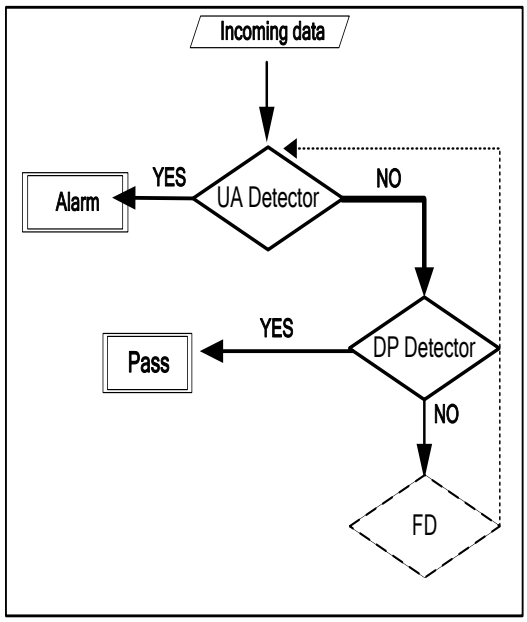

Figure 2. Nest-mate Recognition Approach

Nest-mate recognition approach combines both UA and DP while there is no such combining in nature. This is necessary here to reduce the number of errors in acceptance and rejection and to get the full advantages from them all. Figure 3 below shows the significances of combining between the UA and DP on determining the acceptance and rejection to get an optimal guard.

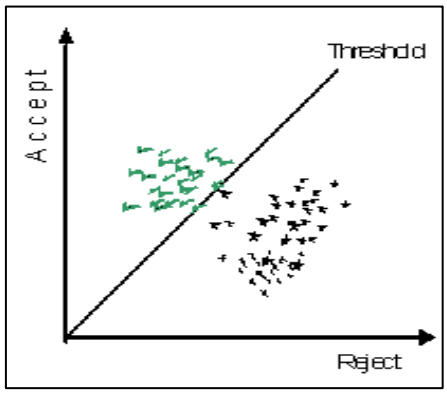

Figure 3. Significance of using an optimal threshold

The third detector in the nest-mate recognition of honeybee approach is a Filtering-Decision *(FD)* which is taking the part of the template updated for *UA*. The packets which have been detected as abnormal and forwarded from *DP* are stored and verified by *FD*. The advantages of this procedure are to make UA detector more effective by updating its classifier with new records (novel attacks) in real-time. Moreover, the *UA* detector is trained during the execution of the system by adding new intrusions online. Subsequently, the *UA* detector will be more accuracy on detecting the type of intrusion or a new untrained attack. Figure 4 shows the overview of FD framework.

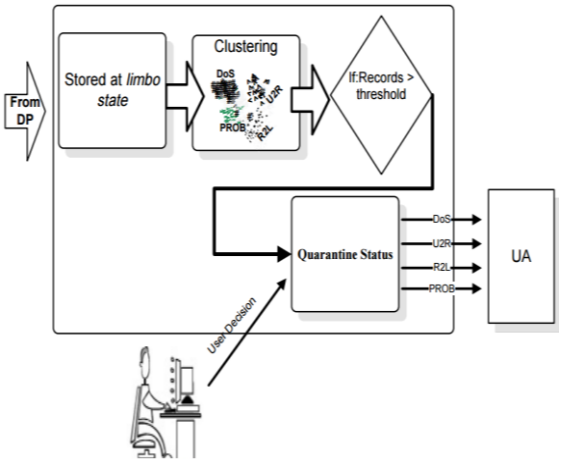

Figure 4. Overview of FD framework.

One of the important requirements for the technique to support the proposed approach is the ability of learning. Besides that, this technique is supposed to distinguish different characteristics after some level of training. Thus the neural network has been chosen to be the main component of the model because of the many features that neural network poses such as the ability of learning, generalizing attributes even with noisy data, and the capability of classifying patterns effectively. These features can be further used to improve detection and reduce false alarms in the intrusion detection system.

After the training phase, the neural network will be able to make the distinction between both normal and anomalous and then within anomalous between different attack classes. Once the neural network is trained, it can be used to classify new data sets whose input/output associations are similar to those that characterize the training data set.

3.1. The Training Components Part

The objective of the training part is to train the neural network such that it becomes perceptive and sensitized to the specified dataset. At first, the

dataset read by the initialization function. Then, the weights of the neural network are generated by the Bees Algorithm training. From the data file and the parameters given by the user, the initialization function will provide the user with random values as weights. The summary of training process illustrated in Figure 5. Once the network is trained, it can be used to classify new data sets whose input/output associations are similar to those that characterize the training data set.

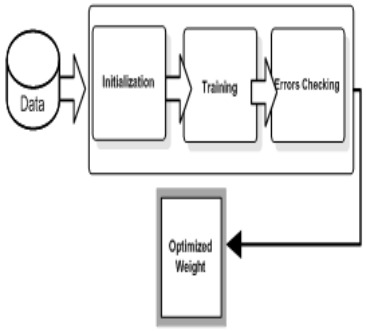

Figure 5. Neural Network Training

3.1.1. Neural Network Training. In the proposed work, the problem and data clearly indicate that the neural network learning is the supervised learning type. The training data task consists of T input-output (vector-valued) data pairs as following:

- Neural Network (NN) consists of a set of neurons or nodes which are interconnected with each other. According to [11], each neuron in the network is able to receive input signals, to process them and to send an output signal. Moreover, each neuron is connected at least with one neuron, and each connection is evaluated by a real number, called the weight coefficient, that reflects the degree of importance of the given connection in the neural network.

$$\boldsymbol{u(n)} = (\boldsymbol{x_1^0}\,(\boldsymbol{n}), \dots, \boldsymbol{x_k^0(n)})^{\boldsymbol{t}}, \boldsymbol{d(n)} = (\boldsymbol{d_1^{k+1}(n)}, \dots, \boldsymbol{d_L^{k+1}(n)})^{\boldsymbol{t}} \quad \dots. 1$$

where $\boldsymbol{n}$ denotes training instance. The output of the neural network is a function of synaptic weights $\boldsymbol{W}$ and input values $\boldsymbol{x}$, $\boldsymbol{i}$.e., $\boldsymbol{Y = f\,(x,W)}$. The ***ith*** neuron can be written as equation 2

$$y_i = f_i\left(\sum_{j=1}^{n} w_{ij}\, x_j + \theta_i\right) \quad \dots. 2$$

Where $\boldsymbol{y_i}$ is the output of the node, $\boldsymbol{x_j}$ is the **jth** input to the node, $\boldsymbol{w_{ij}}$ is the connection weight between the node and input $\boldsymbol{x_j}$, $\boldsymbol{\theta_i}$ is the threshold (or bias) of the node, and $\boldsymbol{f_i}$ is the node transfer function.

$$\boldsymbol{E(w(t)) = \frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K}(d_k - 0_k)^2} \quad \dots. 3$$

where, $\boldsymbol{E(w(t))}$ is the error at the $\boldsymbol{t}$th iteration; $\boldsymbol{w(t)}$, the weights in the connections at the $\boldsymbol{t}$th iteration; $\boldsymbol{d_k}$, the desired output node; $\boldsymbol{0_k}$, the actual value of the $\boldsymbol{k}$th output node;$\boldsymbol{K}$, the number of output nodes; $\boldsymbol{n}$, the number of patterns.

4. Evaluation Criteria

Detection rate and a false positive rate are two main performance indicators. The false positive rate especially is critical to the performance of an intrusion detection system as a small difference of the false positive rate may translate into high number false alarms compared to the actual number of real alarms [10]. In most of the situations, it is not the ability of identifying attacks but rather its ability of suppressing false alarms that limit the performance of an intrusion detection system. The two major indications of performance are illustrated below:

$$\mathbf{DR} = \frac{\textbf{Detected intrusion samples}}{\textbf{Total number of samples}}$$

$$\mathbf{FPR} = \frac{\textbf{Normal samples incorrectly classified as intrusion}}{\textbf{Total number of samples}}$$

We tested each proposed detectors of the nest-mate recognition approach individually in order to evaluate each performance accurately.

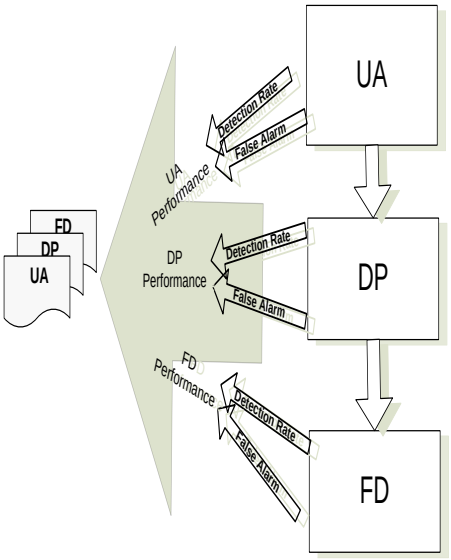

Figure 6. Net-mate Recognition Approach Evaluation

This way reduces the computation required by the system, and facilitates fine tuning and control. Results of detection and false positive rate will be combined from each method or classifier to show the performance of the proposed IDS as shown in the Figure 6.

From Figure 6 we can see that a proposed system starts detecting by UA. The UA detector will classify an instance as *DoS, Probing, R2L, U2R* or 'unknown'. Whereas 'unknown' is further refined to either normal or 'abnormal' at the DP detector.

Finally, the FD detector will refine 'abnormal' further to such of intrusion classification, , according to the four-class taxonomy of [12], each class is further stored and triggered to update the UA detector. Therefore, UA detector will be trained in different way than DP detector based on the task requirements. UA will learn the characteristics of attacks whereas DP will learn characteristics of normal connection.

4.1. UA detector experiment

In this experiment, four types of attack data (PROBE, DoS, R2L, U2R) and normal data were used to test the performance of UA detector. Moreover, the most challenges to UA is to identify the known and unknown intrusions and classify them to their four major classes as most intrusion detection systems fail overcome this task [13].

The result of the first intrusion detector implementation is shown in Table 1.

Table 1. Results of UA detector

	DR (%)	FPR (%)
Dos	99.30	0.77
R2L	89.03	1.70
Probe	98.21	2.17
U2R	93.16	0.90

From Table 1 we can deduce the efficient use of the UA detector. The DoS got the highest detection rate (99.30%) and the lowest false alarm rate (0.30%). Probe is the next higher DR, got (98.21%). This can be explained by the fact that NN network learned more about DoS and Probe during the learning process because of the majority presentation of theirs records in the learning data set (10% KDD). R2L and U2R attack categories are also got a high detection rate (89.03%, 93.16% respectively) but not high as the DoS and Probe, this can be due to the same reason of the lack of their presentations during the learning phase.

4.2. DP Detector Experiments

DP detector is designed based on the desirable characteristics or profile of normal activities. In this experiment, the remaining records which include normal data and some abnormal ones that UA could not capture will flow to DP. Only normal data (attack-free) is used for training the DP. Therefore the DP detector is only recognized whether the packet is normal or abnormal. The abnormal ones which contain attack or suspicious will be followed to the FD. The result of the DP testing is shown below in Table 2.

Table 2. Results of DP detector

	Number of records (Correctly Classified)	Number of records (False Classified)	Detection Rate(%)	FPR (%)
Normal	**62339**	**1730**	**97.30**	**2.30**

Table 2 shows the power of the trained neural network in identifying the unknown intrusion by detecting the deviation of normal. The DP has the overall detection rate of 97.30% and 2.30% false positive rate. We can notice that DP detects more anomalies and intrusion than UA, but at the same time, the DP gets more but a little bit increasing of false alarm. Hence, there is a tradeoff here. More strict condition for a connection to be normal will result in more anomalies and more false alarms. According to [14], anomaly detectors perform better than other detectors over KDD'99 dataset using various machine-learning algorithms. One explanation to this might be due to the complex distribution of training samples and embedded attack patterns in the KDD'99 data [15].

4.3. *FD* Detector Experiment

The FD detector was trained as misuse detector to identify certain well-known intrusions. The records that have been detected as abnormal or flagged as novel attacks were forwarded from *DP* and stored to be verified by *FD*. There is a connection between FD and UA to modify the later with novel intrusions that have been detected and clustered.

The result of FD is shown below in Table 3. It shows that the FD has both a high detection rate and low false positive rate.

From Table 3 it is noticeable that the flowed data in the testing stage is distributed into the four types of major attacks. These attacks are recognized as novel intrusions as UA and DP couldn't detected. The FD detected each intrusion and referred it to its major class.

Table 3. The Experiments Result of FD

	Data	DoS	PROBE	U2R	R2L	DetectionRate	FalsePositive
DoS	flowed identified	554 542	0 6	0	0	542/554 = 97.8%	6/550 = 0.01%
PROBE	flowed identified	3	100 93			93/100 = 93%	3/100 = 0.03%
U2R	flowed identified	2	1	50 45	1	45/50 = 92%	4/50 = 0.08%
R2L	flowed identified	1			25 23	22/25 = 88%	3/25 = 0.12%
Remain at limbo state		6	4	1	0		

It is important to notice that many predictions seem not accurately. The practical reason is that the number of training instances is small. Alternatively, there are many novel intrusions that detected where as they did not appear during the training phase. Another important point to note is that every classifier is connected to UA classifiers in order to modify the UA structures with the novel intrusions.

5. Using NSL-KDD_2009 to test the proposed approach

The new data set, NSL-KDD as suggested by [16], which consists of selected records of the complete KDD dataset is using to test the proposed approach. The dataset is publicly available for researchers and has advantages over the original KDD data set.

Table 4. Experimental result of NSL-KDD dataset test

Record Type	No. of Patch	No. of Detection Records		FN	FP
		UA	DP		
NSL-KDD	1st_Patch= 1000 records	620	330	20	30
	2nd_Patch= 1000 records	407	593	0	0
	3rd_Patch= 1000 records	498	489	7	6
	4th_Patch= 1000 records	795	200	0	5
	5th_Patch= 1000 records	962	38	0	0
	6th_Patch= 1000 records	169	820	4	7
	7th_Patch= 1000 records	823	177	0	0
	8th_Patch= 1000 records	338	659	1	2
	9th_Patch= 1000 records	572	421	3	4
	10th_Patch= 1000 records	619	380	0	1
Overall		5803	4107	35	55
The Overall Rate		99.1%		0.35 %	0.55%

The new dataset can be applied as an effective benchmark data set to help researchers to compare different intrusion detection methods [17]. The generated data sets, KDDTrain+ and KDDTest+, included 125,973 and 22,544 records, respectively. A 20% subset of the KDDTrain+.txt file is used for training the proposed IDS system whereas a subset of the KDDTest+.txt file is used for the testing phase. Table 4 shows the overall results on the NSL-KDD dataset.

Table 4 illustrates the high performance of the proposed IDS. It shows the higher detection rate 99.1% and a low False Positive Rate 0.55% and False Negative Rate 0.35% of the system performance. The results obtained in this test demonstrate clearly the benefit of the proposed approach on the NSL-KDD dataset.

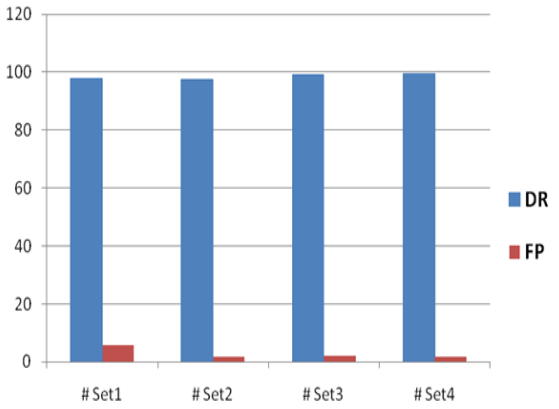

Figure 7. The Experimental results from initial population testing compared to DR and FPR

More specifically, it can be observed that UA detector is indeed capable of detecting more than half of the intrusions either new or old whilst the task of DP detector is efficiently demonstrated; it is obvious that most of the undetectable intrusions by UA are detected by DP detector. In practice, the DP detector is more sensitive and restrictive if found any variation from normal data. The combined of UA and DP detectors in proposed approach leads to get high detection rate and low false alarm. Following graph gives the overall evaluation over the initial population testing on different subset.

6. Result from Specific Population Testing

In this experiment, the performance measurement of proposed IDS is tested with specific population testing. The attacks in the data set fall into four main categories: DoS, R2L, U2R, and PROBE. In order to demonstrate the abilities of detecting different kinds of intrusions, the training data and testing data cover all intrusion categories. Totally, 1,200 attack data and 1,000 normal data were prepared for training and another set of 1,200 attack instances and 1,000 normal data were selected as the testing data. The attack population data are selected according to the measure attack categories and have the same approximate distribution as the KDD dataset. The selected data records are illustrated in Table 5.

Table 5. Initial population testing of KDD

Attack	Attack Name	Records	Total
Normal		1000	100
DoS	Neptune	155	517
DoS	Smurf	174	
DoS	Back	92	
DoS	Land	40	
DoS	Apache2	33	
DoS	Teardrop	23	
Probe	Ipsweep	129	369
Probe	Nmap	59	
Probe	Portsweep	77	
Probe	Satan	44	
Probe	Mscan	36	
Probe	Saint	24	
U2R	buffer_overflow	82	217
U2R	sqlattack	79	
U2R	Perl	8	
U2R	Xterm	22	
U2R	Rootkit	26	
R2L	guess_passwd	41	97
R2L	Imap	2	
R2L	ftp_write	22	
R2L	Phf	20	
R2L	Sendmail	12	

In the experiment, the performance measure of UA and DP are carried out solely on the selected data subset from the corrected.gz file of the KDD'99 dataset which contains test data with corrected labels and other attacks examples from 10% KDD. The primarily results show that it is possible to increase the detection rate and reduce false alerts.

Each detector in honeybee approach has a good performance in identifying intrusion patterns and detects attacks. Table 6 shows the experiment results.

The results show that UA & *DP* detectors have high *Detection Rate* and low *False Positive* even with small data set. This observation leads to consider that duplicates may also lead to somewhat deceptive results during testing, since the ability to detect one instance will be multiplied according to the number of duplicates. The proposed approach demonstrates better performances in the most number of attacks categories and less false alarm. Based on the results that shown in previous tables, it can be seen that the proposed approach has a good performance for detecting intrusion in computer networks.

Table 6. Experimental result from selected population testing

Record Type	No. of Records	No. of Detection Records			
		UA	DP	DR %	False Alarm
Normal	1000	17	963	*963/1000= 96%*	17/1000=*1.7%*(FP)
Probe	369	202	165	*367/369= 99%*	2/369=*0.5%*(FN)
DoS	517	328	188	*516/517= 99.8%*	1/517=*0.19%*(FN)
U2R	217	82	134	*216/217= 99%*	1/217=0.46%(FN)
R2L	97	22	73	*95/97= 98%*	2/97= 2.1%(FN)

Moreover, the overall result of the detection of old and new attacks in different classes are high.

7. Conclusion

This paper defined a problem statement based on existing research of IDS and tried to match the solution from nature, specifically from nest-mate recognition of honeybee. This paper shows detection deficiency of IDS detector and demonstrates the value of the proposed approach in terms of how it can be used to reduce false alerts and increase the detection accuracy. Moreover the proposed IDS have been tested over different datasets and partitions of datasets.

This study provides guidance to future initiatives to emulate the models of intrusion prevention of honeybee colony in nature which may produce fruit knowledge that could be applied practically to IPS.

8. References

[1] ISACA, and RSA Conference, “State of Cybersecurity: Implications for 2015”. http://www.isaca.org/cyber/ Documents/State-of-Cybersecurity_Res_Eng_0415.pdf. 2016.

[2] Lin, Wei-Chao, Shih-Wen Ke, and Chih-Fong Tsai. "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors". *Knowledge-based systems* 78 (2015): 13-21.

[3] Jan N.Y., Lin S.C., Tseng S.S. and Lin N.P., (2009). A decision support system for constructing an alert classification model, Expert Systems with Applications 36, pp. 11145–11155.

[4] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Intelligence", NY: Oxford University Press, NewYork, 1999.

[5] Ali, G.A., Jantan, A., Ali, A.: Honeybee-Based Model to Detect Intrusion. In: Park, J.H., Chen, H.-H., Atiquzzaman, M., Lee, C., Kim, T.-h., Yeo, S.-S. (eds.) ISA 2009. LNCS, vol. 5576, pp. 598–607. Springer, Heidelberg (2009).

[6] Yang C, Jie Chen J, Tu X., (2007a). Algorithm of fast marriage in honey bees optimization and convergence analysis. In: IEEE international conference on automation and logistics, Jinan, pp 1794–1799.

[7] Cathey, R., et al. "Misuse detection for information retrieval systems."*Proceedings of the twelfth international conference on Information and knowledge management*. ACM, 2003.

[8] Wu, Shelly Xiaonan, and Wolfgang Banzhaf. "The use of computational intelligence in intrusion detection systems: A review." Applied Soft Computing 10.1 (2010): 1-35.

[9] Sekar, R., et al. "Specification-based anomaly detection: a new approach for detecting network intrusions." Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002.

[10] Wilson, R., and Obimbo, C., “Improvements on Self-Organizing Feature Maps for User-to-Root and Remote-to-Local Network Intrusion Detection on the 1999 KDD Cup Dataset”, International Journal for Information Security Research (IJISR), Volume 2, Issue 2, June 2012.

[11] Ibrahim, Laheeb M., Dujan T. Basheer, and Mahmod S. Mahmod. "A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self-organization map (SOM) artificial neural network." Journal of Engineering Science and Technology 8.1 (2013): 107-119.

[12] Kendall, K., “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, M. Eng. Thesis” Massachusetts Institute of Technology, Massachusetts, United States, June 1999.

[13]Ghorbani, Ali A., Wei Lu, and Mahbod Tavallaee. *Network intrusion detection and prevention: concepts and techniques*. Vol. 47. Springer Science & Business Media, 2009.

[14] Toosi, Adel Nadjaran, and Mohsen Kahani. "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers." Computer communications 30.10 (2007): 2201-2212.

[15] Song, H., and Lockwood, J.W., "Efficient packet classification for network intrusion detection using FPGA." *Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays*. ACM, 2005.

[16] Tavallaee M., Bagheri E., Lu W., and Ghorbani A., (2009). A Detailed Analysis of the KDD CUP 99 Data Set. Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA).

[17] Ghorbani, A.A., Lu, W., Tavallaee, M., (2009). Network Intrusion Detection and Prevention, Springer US. Doi: 10.1007/978-0- 387-88771-5_7.

