ICIT 2015 The 7th International Conference on Information Technology
doi:10.15849/icit.2015.0044

Enhancing Intrusion Detection System (IDS) by Using Honeybee Concepts and Framework

Ghassan Ahmed Ali
College of Computer Science and Information System
Najran University
Najran, Kingdom of Saudi Arabia
gaabdulhabeb@nu.edu.sa

***Abstract*— Intrusion Detection System has been studied for more than ten years. Though Artificial Intelligence (AI) techniques have been integrated to improve IDS but the success is still far from satisfaction. Thus, we believe a new strategy to improve IDS is badly needed. One of the solutions is by imitating the honeybee colony that can successfully protect their colony. In fact, the honeybee colony system and problems are quite similar with network system, and the way the bees protecting their colony can also be considered similar with the IDS in the network system. In this paper, we investigated the honeybee colony system as well as their detection system to get improvement methods for IDS engine in order to enhance IDS system for a better network defense. The adaptation of the honeybee protection and defense system itself is a new knowledge that can help other systems such as antivirus, antimalware, or even defense system to imitate the AI techniques in performing their functions. We train the proposed system with different types of attacks data and model different types of attack signatures. The performance of the proposed IDS is evaluated using NSL-KDD data set. The experiments show that the performance of the proposed model can detect novel intrusions and reduce false alarms.**

***Keywords*— honeybee; intrusion detection; system Security**

I. INTRODUCTION

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 [1]. 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 Bonabeau et al. [2]. 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 [4].From these models there being extracted many features were being utilized by engineering and computer science [5].

Despite the strength of security system of honeybee behavior in nature (such as guarding, perception, information flowing, nest policy and rules, etc), however, it is still "raw material" in computer sciences application. Previous research in biology has shown that honeybee guard made very few errors in accepting nest-mates and rejecting non-nest-mate [6]. In addition, Honey bees use an early-warning system to detect threats and defend the nest [7]. The multilayer protection in honeybee colony and the diversity of defenses can be viewed as a distributed detection system. All these features in the behavior of the honeybees can be a construction of a novel security model to develop the accuracy of IDS.

Honeybees in nature survive in difficult environments, different levels of threats to security. These threats motivate bees to be able to detect and respond quickly on any aggressive acts that may attack the colony [8]. This paper focuses on how bees solve such security problems regarding the detection to crossover directly to IDS. This can be achieved through the use of the approach and architecture that are based on honeybee mechanisms. The investigation of this approach yield new insight into computer sciences.

This paper investigates a new method in the direction of construct a significant decision to accept or reject the incoming packets based on packet characteristic that each packet posses, in addition to get accurate decision to accept valid packets and reject intruder. This development for packet classification will improve robustness and accurate detection.

doi:10.15849/icit.2015.0044

II. THE HONEYBEE GUARD APPROACH AND DETECTOR COMPONENT

The mechanisms 'D-present/U-absent' which is used to match the characteristics among individuals and rejecting non-group members in nature is proposed as a model by [9]. This model assumes that recognition system of nest-mates is detecting either the presence or the absence of the characteristics they carry.

In this paper, the methods Undesirable-Absent (UA) and Desirable–Present (DP) that honeybee guard uses in nature in order to filter the incomer are applied to IDS. Undesirable-Absent (UA) calculates the undesirable characteristics that found in a receiving packet and compares it with the internal characteristics template. In order to apply the idea of undesirable-absent in the domain of intrusion detection, we need to determine the characteristics that will represent the malicious or attacks (the non-nestmate in real honeybee). For this reason, the dataset collected by DARPA 1999 and preprocessed for the KDD '99 competition have several relevant features that can be used as characteristics for the attack properties. Neural network will be trained to recognize the characteristics of attacks in order to classify these characteristics as undesirable characteristics during the testing phase.

The Desirable-Present detector compares between the characteristics of the forwarded packet and its "template" which contains the desirable characteristics of an accepted packet. The Desirable-Present detector built of normal data. The normal of "10% KDD" Cup 1999 dataset, which is free from attacks used to train the Desirable-Present detector in order to recognize the desirable characteristics of the incoming connection records. The advantage is to aim the Desirable-Present detector to detect new types of intrusions; as unexpected intrusions are deviating from normal network.

After preprocessing the data and training the neural network, the task would be to determine whether the test data belongs to normal or abnormal based on the features of connection records from a given new test data. The result of this learning process is a neural network which is capable of detecting anomalies in the traffic during the testing phase "corrected KDD".

The proposed IDS is divided into three main modules. Practically, each module is implemented to perform a designated intrusion detection task. Moreover, the generality of the detector is ensured by the standard data representation schemes for input/output adopted by the constituent modules.

III. STRUCTURE OF THE PROPOSED IDS

The core components of the detector modules consist of a set of soft computing classifier to have the ability to detect both well-known and novel intrusions. Figure 1 shows general structure of the proposed system. The description and the interactions of the main modules are as following:

- Training Data Processing: A file called "Training Set" is input to the IDS. The file contains network data from the KDD Cup 1999 intrusion detection data set. Each row of the file contains an instance of the data, and each column represent unique attributes. The data also can be presented directly from the live network dumped from any sniffer. The task handled by the data processing module is to normalise or cleanses the given dataset for the data mining. Since the performance of any IDS not only depends on output of the IDS but also on input traffic

- Data mining: This module represents the data mining techniques, and uses the training data to train the system. The attributes of the trained data mining are stored for later use, during the testing. The output of the data mining module is a text file containing the parameters and the weight of the learnt neural network. This approach has the advantage of being able to automatically retrain intrusion detection models on different input data that include new types of attacks. The training processes are further explained in the coming sections.

- IDS Testing: In this phase, the data mining will be validated to ensure its usefulness. In order to prove that the proposed system is not only successful on the training set, a separate test set with new data is used to test the system. This data also comes from the KDD Cup 1999 set. Typically, network based IDS process system activities based on network data and make a decision to evaluate the probability of action of these data to decide whether these activities are normal or intrusion.

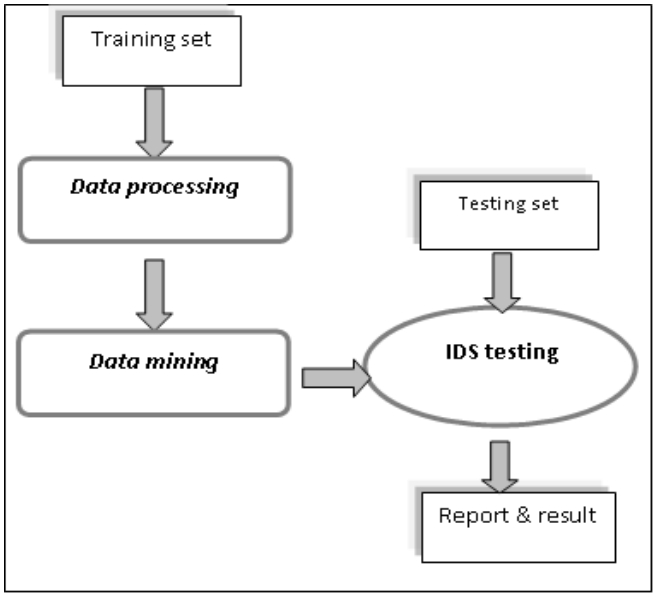

Figure 1: Summary of the Proposed IDS Structure.

doi:10.15849/icit.2015.0044

In this study, we use Artificial Intelligence (AI) techniques in order to take the advantages of the new approach to improve the IDS. According to [10], the concept of using AI to solve the two IDS problems is very efficient. The generalization of AI makes possible decreases of false alarms as well as increases the accuracy of an intrusion detection process.

One of the important requirements for the technique to support the proposed approach is the ability of learning. Beside 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.

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. The training components train the neural network such that the internal structure or topology of the given dataset. At first, the data set 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 2. 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.

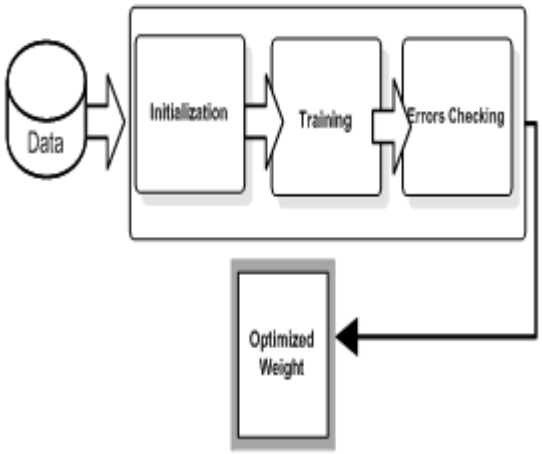

Figure 2: Neural Network Training

A. 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:

The 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

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

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.

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

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.

doi:10.15849/icit.2015.0044

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%

2. 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 [1]. 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:

DR = detected intrusion samples / total number of samples ***(4)***

FPR = normal samples incorrectly classified as intrusion / total number of samples ***(5)***

3. USING NSL-KDD_2009 TO TEST THE PROPOSED APPROACH

The new data set, NSL-KDD as suggested by [12], which consists of selected records of the complete KDD data set is using to test the proposed approach. The data set is publicly available for researchers and has advantages over the original KDD data set.

The new dataset can be applied as an effective benchmark data set to help researchers to compare different intrusion detection methods [13]. 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 2 shows the overall results on the NSL-KDD dataset.

Table 2: Experimental Result in Test NSL-KDD Dataset.

Table 2 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. More specifically, it can be observed that Undesirable-Absent detector is indeed capable of detecting more than half of the intrusions either new or old whilst the task of Desirable-Present detector is efficiently demonstrated; it is obvious that most of the undetectable intrusions by Undesirable-Absent are detected by Desirable-Present detector. In practice, the Desirable-Present detector is more sensitive and restrictive if found any variation from normal data. The combined of Undesirable-Absent and Desirable-Present detectors in proposed approach leads to get high detection rate and low false alarm.

Result from Specific Population Testing

In this experiment, the performance measure 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 data set. The selected data records are illustrated in Table 3 below.

Attack Category	**Attack Name**	**Records**	**Total**
Normal		1000	1000
DoS	Neptune	155	517
DoS	Smurf	174	
DoS	Back	92	
DoS	Land	40	
DoS	Apache2	33	
DoS	Teardrop	23	
Probe	Ipsweep	129	369

doi:10.15849/icit.2015.0044

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	

Table 3: Experimental Result from Initial Population Testing

In the experiment, the performance measure of *Undesirable-Absent* and *Desirable-Present* detector are carried out solely on the selected data subset from the corrected 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's possible to increase the detection rate and reduce false alerts. Each method in honeybee approach has a good performance in identifying intrusion patterns and detects attacks. Table 4 shows the experiment results. The results show that *Undesirable-Absent* & *Desirable-Present* detectors have high *Detection Rate* and low *False Positive* even with small data set

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)

Table 4: Experimental Result from Selected Population Testing

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

IV. CONCLUSION

The focus of this paper was to demonstrate how productive the crossover between biology and computer science can be. The detection system in honeybee, which keeps the colony safe, was the basis frame of the research to improve the effectiveness of IDS. The new approach is used to improve the IDSs at the detector level to distinguish between the innocuous and intruders using the way that honeybee is used in nature. Characterizing the incoming packets to support detection was significant. Characterization methods have ranged using trained neural network that it becomes perceptive and sensitized to detect intrusions.

To examine the feasibility of our approach, we conducted several experiments. The experimental results demonstrate that the proposed approach can improve the detection deficiency issue by reducing the false alerts and increasing the detection accuracy.

REFERENCES

[1] 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.

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

[3] 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)

[4] 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

[5] Ali, Ghassan Ahmed and Jantan, Aman: A New Approach Based on Honeybee to Improve Intrusion Detection System Using Neural Network and Bees Algorithm. Springer Berlin Heidelberg. 2011. http://dx.doi.org/10.1007/978-3-642-22203-0_65

[6] Grüter C, Kärcher MH, Ratnieks FLW (2011). The Natural History of Nest Defence in a Stingless Bee, Tetragonisca angustula (Latreille) (Hymenoptera: Apidae), with Two Distinct Types of Entrance Guards. Neotrop. entomol. http://dx.doi.org/10.1590/S1519-566X2011000100008.

[7] Breed, D., Guzmán-Novoa, E., Hunt, G.J.: Defensive behavior of honey bees:organization, genetics, and comparisons with other Bees. Annual Review of Entomology 49, 271–298 (2004)

[8] Couvillon, M. J., & Ratnieks, F. L. W. (2007). Odour transfer in stingless bee marmelada (Frieseomelitta varia) demonstrates that

doi:10.15849/icit.2015.0044

entrance guards use an "undesirable–absent" recognition system. Behavioral Ecology and Sociobiology, 62(7), 1099-1105. Springer.

[9] Sherman P.W., Reeve H.K., Pfennig D.W. Recognition systems. In Behavioural ecology: an evolutionary approach KrebsJ.R., DaviesN.B. 1997pp. 69–96. Eds. Oxford, UK:Blackwell Science.

[10] Servin A. and Kudenko D. (2008). Multi-agent reinforcement learning for intrusion detection, lecture notes in computer science, vol. 4865; 2008. p. 211–23.

[11] Daniel Svozil, Vladimir Kvasnicka, Jiri Pospichal (1997). Introduction to multi-layer feed-forward neural networks, Chemometrics and Intelligent Laboratory Systems, Volume 39, Issue 1, November, Pages 43-62, ISSN 0169-7439, DOI: 10.1016/S0169-7439(97)00061-0.

[12] 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).

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