I. J. Computer Network and Information Security, **2018, 11, 42-49**
Published Online November 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2018.11.05

Cuckoo Optimisation based Intrusion Detection System for Cloud Computing

D. Asir Antony Gnana Singh
Anna University, BIT-Campus, Tiruchirappalli, Tamil Nadu, India
E-mail: asirantony@gmail.com

R. Priyadharshini, E. Jebamalar Leavline
Anna University, BIT-Campus, Tiruchirappalli, Tamil Nadu, India
E-mail: dharshinipriya245@gmail.com, jebilee@gmail.com

Received: 04 June 2018; Accepted: 16 September 2018; Published: 08 November 2018

***Abstract*—**In the digital era, cloud computing plays a significant role in scalable resource sharing to carry out seamless computing and information sharing. Securing the data, resources, applications and infrastructure of the cloud is a challenging task among the researchers. To secure the cloud, cloud security controls are deployed in the cloud computing environment. The cloud security controls are roughly classified as deterrent controls, preventive controls, detective controls and corrective controls. Among these, detective controls are significantly contributing for cloud security by detecting the possible intrusions to prevent the cloud environment from the possible attacks. This detective control mechanism is established using intrusion detection system (IDS). The detecting accuracy of the IDS greatly depends on the network traffic data that is employed to develop the IDS using machine-learning algorithm. Hence, this paper proposed a cuckoo optimisation-based method to preprocess the network traffic data for improving the detection accuracy of the IDS for cloud security. The performance of the proposed algorithm is compared with the existing algorithms, and it is identified that the proposed algorithm performs better than the other algorithms compared.

***Index Terms*—**Intrusion detection system, Cloud security, Cloud computing, Feature selection, Machine-learning algorithm.

I. Introduction

In recent days, it is impossible to imagine the computing and information technology without cloud computing. Cloud computing plays a major role in computing, information and resource sharing with its effective services such as software as a service, platform as a service, infrastructure as a service and so on. Securing the cloud is a challenging task among the researches as it is a dynamic and multi-tenet environment. To provide cloud security, different types of cloud security controls are deployed to prevent the possible attacks or to reduce the intensity of the attacks from intruders. The cloud security controls are classified into four categories namely deterrent controls, preventive controls, detective controls and corrective controls. The deterrent control mechanisms are employed to reduce the level of attack by providing an alert.

The preventive control methods strengthen the preventive actions against the threat or attack. Corrective controls decrease the severity of the attack. Detective controls predict and identify the possible attacks and intimate those to network administrator to avoid the attack [1]. The intrusion detection system (IDS) is a category of detective controls in the security control of the cloud environment. The IDS is employed in the cloud environment to predict and detect the suspicious, malicious, abnormal, attacker and intruder data packets or network flow and intimate them to the administrator to prevent the possible attacks. On other hand, the firewall is employed for the network security in cloud environment. However, the firewall only restricts or drops the data packets that violate the organisational or network policies. Moreover, the intruders pertain and pass through the firewall on the network. Therefore, the firewall fails to provide security for all possible vulnerable attacks. Hence, intrusion detection is used to detect the intruders in the cloud environment.

The IDSs are categorised into two based on where it is located in the cloud environment as network IDS (NIDS) and host-based IDS (HIDS). The NIDS is placed in the network where the cloud is connected with the internet to provide the service to the cloud users as shown in Fig. 1. The NIDS scans the network flow data and detect the data packet whether the packet comes from the intruders for the possible attack or normal data packet. If the packet comes from the intruders with the intension of attack, then alert message is given to the network administrator to take preventive action against that attack. Thus, the NIDS safeguards the network.

The HIDS is placed in the each host that is deployed in the cloud as shown in Fig. 1. The HIDS monitors the data packet that comes to the host machine and predicts or detects the packet that comes from the intruders. If the packet comes from the intruders, it gives the alert message to the system administrator to take preventive action against the possible attack.

Furthermore, the IDSs are classified into two based on their working principle, namely signature-based IDS and anomaly-based IDS. In signature-based IDS, the pattern of the packet and signatures are used to identify the packet from the intruders. In this approach, the accuracy of the detection is high. However, signature-based IDS method fails to detect the packet when signature of the arriving packet is not available with the system. In other words, it fails to detect the packets with unknown pattern or signature. On the other hand, the anomaly-based IDS is developed with the known possible attack data or network traffic flow data using any one of the machine-learning algorithms. This approach can detect the intruder packets even with unknown signature or pattern.

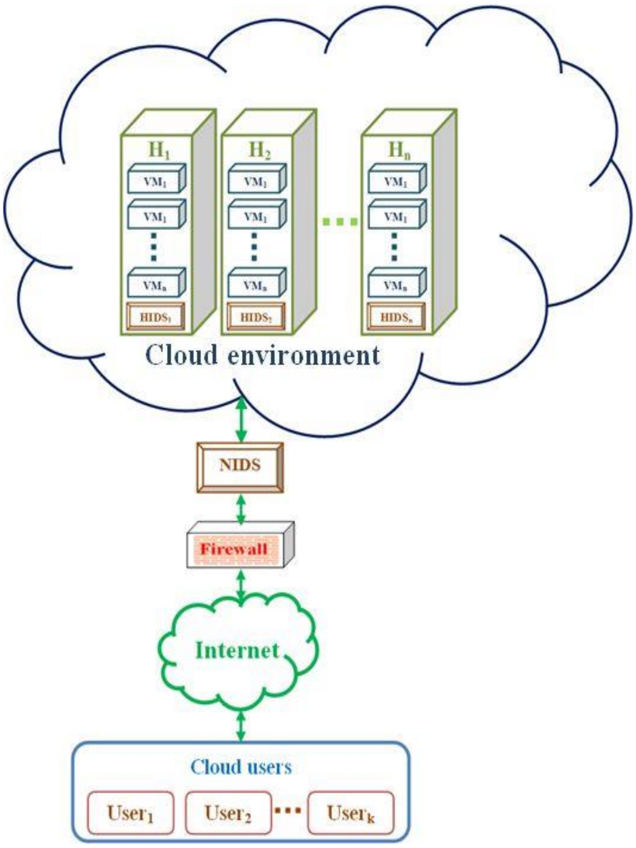

Fig.1. Intrusion Detection in Cloud Environment

However, in anomaly-based method, the false positive (FP) rate of intrusion detection is higher than the signature-based approach. To overcome this, the data preprocessing method is necessarily adopted to develop IDS using machine-learning algorithm [2]. Moreover, FP rate reduces the accuracy of IDS in detecting the intruder packet. If the IDS wrongly detects the packet, it can lead to vulnerable possible attacks and threats in the cloud environment. Hence, improving the accuracy of the IDS is needed to secure the cloud computing environment. Therefore, this paper presents an anomaly-based IDS using cuckoo optimisation-based data-preprocessing technique for cloud security.

The reset of the paper is organised as follows: Section 2 reviews the literature. Section 3 explores cuckoo optimisation-based IDS (COIDS). Section 4 discusses the implementation and experimental results. Section 5 concludes this paper.

II. Literature Review

This section reviews the research works that are related to the proposed work. In general, the anomaly-based IDSs are developed using machine-learning algorithms such as classification algorithm. Initially, the data are collected from the network flow or from the network log data. Then, this data is given to the classification algorithm. The classification algorithm learns the network data and develops the classification model. This classification model is known as intrusion detection model. It is used to detect the anomaly packets that travel through the network or flow into the host. To improve the accuracy of the classification model, the researchers use data-preprocessing technique that is known as feature selection or variable selection. The feature selection is a process of removing the redundant and irrelevant feature from the network data. Hence, most of the researchers employed the data preprocessing technique such as feature selection to improve the accuracy of the IDS.

Al-Jarrah et al. presented an IDS with data preprocessing and machine-learning approach to improve the accuracy of the IDS [3]. Ambusaidi et al. proposed a filter-based feature selection to improve the accuracy in intrusion detection [4]. El-Khatib used the information gain (IG) ratio measure and the k-means classifier to select the optimal significant features from the data set to improve the accuracy of IDS [5]. Moreover, Mishra et al. presented a study on intrusion detection in cloud environment [6]. Viegas et al. proposed a feature-selection method to improve the accuracy of IDS and to reduce the energy consumption in embedded system [7].

Feature selection is employed in the development of IDS to improve detection accuracy of the IDS. Feature selection can be categorised into three types, namely filter, wrapper and embedded methods. Besides improving the accuracy of the IDS, feature selection is also used for various pattern-recognition applications to improve their recognition or detection accuracy. Mistry presented a feature-selection method with genetic algorithm (GA) and particle-swarm optimisation (PSO) algorithm for facial emotion recognition [8]. Lagrange suggested a feature-selection method for classifying the remote sensing images [9]. Gülsen and Ta skın presented a feature-selection method for classifying the hyper-spectral images [10]. Wang et al. proposed a feature-selection method with PSO to remove the irrelevant and redundant features from the high dimensional space to improve the accuracy of the classification algorithm [11]. Ma et al. presented a wrapper-based feature-selection method using IG ratio measure with support vector machine and machine-learning algorithm to select the significant features from the data set to improve the accuracy in classification [12]. Huda et al. developed a feature-selection method to improve the accuracy in classification [13]. Nguyen et al. presented a feature-selection method with fuzzy clustering method [14]. Abedinia et al. designed a filter-wrapper approach for selecting the significant features by eliminating the irrelevant and redundant features for forecasting the load and price in electrical power systems [15].

Moreover, many researchers have developed IDS for different computing and network environments. Yang presented an IDS for supervisory control and data acquisition network [16]. Marchang et al. designed an IDS for mobile ad hoc networks [17]. Ha et al. presented an IDS for detecting the suspicious flow in the network [18]. Zhou et al. suggested an anomaly-based IDS for industrial process automation [19]. Lo et al. developed an IDS for smart grid [20].

Furthermore, the wrapper-based feature-selection method produces higher accuracy for the specific task [12, 21]. Hence, the wrapper-based feature selection approach is preferred by the researchers where the recognition task is predefined. The wrapper-based approach needs a searching algorithm to generate the feature subsets. Then, these generated subsets are evaluated using any one of the machine-learning algorithms to select a significant feature subset among them. The traditional searching algorithms take more time to complete the search space, and they fail to produce the optimal solution.

To avoid these limitations, many researchers use the naturally inspired optimisation-based searching techniques to generate feature subsets in feature selection process. Thus, many researches use the naturally inspired optimisation-based searching techniques such as GA-based optimisation [22], PSO [23], differential evolution-based optimisation [24], artificial immune system-based optimisation [25] and ant-colony optimisation [26] in the feature-selection process. The cuckoo search also is a nature-inspired optimisation algorithm that is used in many applications to get optimal solutions [27–29], and this algorithm is used for feature selection. Kulshestha et al. presented a cuckoo search-based feature selection [30].

From this literature, it is observed that the intrusion detection technique is employed for various applications in the computing and network environments. Moreover, the anomaly-based IDS plays a significant role in cloud computing to provide the security. The anomaly-based IDS is developed using machine-learning algorithm. The accuracy of IDS is improved using the data-preprocessing technique such as feature selection. Hence, this paper presents an intrusion detection model that is developed using the machine-learning algorithm such as naïve Bayes classifier. The accuracy of this intrusion detection model is improved by using the proposed wrapper-based cuckoo optimisation-based feature selection (COFS).

III. Cuckoo Optimisation-Based Intrusion Detection System (Coids)

This section presents COIDS. First we present the cuckoo optimisation followed by COFS. Then, we discuss the function of COIDS.

A. FullCuckoo Optimisation (CO) Algorithm

The cuckoo optimisation is derived from the behaviour of cuckoo breeding. Algorithm 1 describes the cuckoo optimisation algorithm. Fig. 2 shows the flowchart representation of cuckoo-optimisation algorithm. This optimisation technique is employed to select the significant features from the data set.

Algorithm 1: CO Algorithm

Input: E and k; // E represents the set of total available eggs and E={e_1,e_2,e_3,…,e_n} where n is total number of available eggs, k represent the size of the host nest HN where HN can hold k number of eggs.

Output: SE // SE set of selected significant k number of eggs with HN.

```
for ($i$ = 0; $i$ < = $n$; $i$ ++)
{
        Find the objective value $\eta_i$ of the each egg $e_i$ using the objective function O($e_i$);
 // Calculate the objective value $\eta_i$ of each egg $e_i$ and $n$ is total number of available eggs.
}
endfor
while (stopping criteria not met)
    {
        Select $k$ number of best eggs based on their objective value $\eta_i$ from $E$ and form the subset SE then place ES into host nest HN;
            // SE contains $k$ number best eggs based on their objective value $\eta_i$ and SE$\subset E$.
        Evaluate the fitness value $\beta_i$= Fv (HN);
        // Evaluate the fitness value $\beta_i$ of HN and HN contains SE
        if (fitness value $\beta_i$ satisfy the stopping criterion)
            {
                Rank and select the higher value of $\beta_i$ of the iterations and select its corresponding SE$_i$ as the set of selected significant eggs and terminate the loop;
}
endif
else
     Update $\eta_i$ of each egg $e_i$ of SE with the abandoned rate $\rho$.
 }
```

B. Cuckoo Optimisation-Based Feature Selection (COFS)

The analogy between the biological terminologies of cuckoo optimisation algorithm and their equivalent feature selection terminologies is tabulated in Table 1. The following assumptions are made to carry out feature selection process for IDS:

(1) The eggs are considered as the features.
(2) The host nests are considered as generated feature subset.
(3) The objective function for each egg is considered

as the calculation of IG for each feature with respect to the class attribute of the network dataset (ND).

(4) The eggs are placed into the host nest based on their objective value (quality) that is considered as the feature set that are formed on the basis of their IG value.

(5) Then the host nest (the nest that contains a set of eggs) is evaluated with the fitness value using fitness function that is considered as the detection accuracy of a feature set formed using na ïve Bayes classification algorithm.

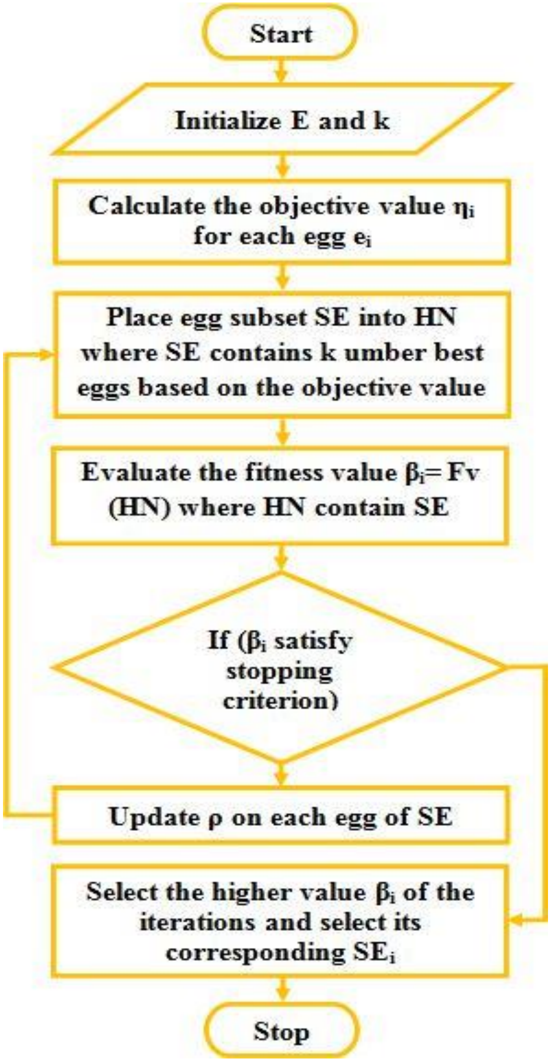
Start
Initialize E and k
Calculate the objective value η_i for each egg e_i
Place egg subset SE into HN where SE contains k umber best eggs based on the objective value
Evaluate the fitness value β_i= Fv (HN) where HN contain SE
If (β_i satisfy stopping criterion)
Update ρ on each egg of SE
Select the higher value β_i of the iterations and select its corresponding SE_i
Stop

Fig.2. Flowchart Representation of CO

If the fitness value satisfies the stopping criteria, rank and select the higher fitness value over the iterations and select the eggs in the corresponding host nest as the set of selected significant eggs and terminate the loop. This means that rank and select the higher detection accuracy over iterations and select the corresponding feature set to build the model. If the fitness value is not satisfied, meaning that the host nest (eggs/feature set) is not suitable for breeding. Then the host nest is collapsed and the objective value of each egg (IG value of each feature of the formed feature set) that is held by host nest is updated with abandoned rate ρ. In other words, the quality of the eggs present in the collapsed host nest is reduced with the abandoned rate ρ. The process is repeated with the assumption (4) until the fitness function satisfies the stopping criteria. The stopping criterion is set as there is no further progress in the fitness value for the iterations.

Table 1. Analogy of Terminologies

Biological Terminology	Equivalent Feature Selection Terminology
Set of total available eggs E={e_1,e_2,e_3,…,e_n} where e represents the egg and n is total number of available eggs	Total feature set F={f_1,f_2,f_3,…,f_n} where f represent the feature and n is the total number of features presents in a network data set ND
HN represents the host nest and k represents the size of HN (i.e. HN can hold only k number of eggs; in other words, totally k number of eggs can be selected at a time)	SF represents the feature subset of F and k represent the size of SF (i.e. SF contains k number of features; in other words, totally k number of features can be selected at a time)
η_i = $O(e_i)$ where $O(e_i)$ represents the objective function of e_i and η_i represent the objective value of e_i (η_i defines the quality of egg)	φ_i = IG (f_i, C) where IG represent the information gain of feature f_i and C represent the class attribute of ND [Equations (1)–(3)]
SE⊂E where SE is the set of eggs which contains k number of eggs that is placed in HN, k is the size of HN	SF⊂F where SF set of features that contains k number of features
β_i = Fv(HN) where β_i is the fitness value and Fv is the fitness function of HN that contain SE	ω_i = da (SF,C) where ω_i represent the detection accuracy and da (.) represents the detection accuracy function [Equation (4)]
ρ is the abandoned rate = 0.05	ρ is the degraded rate = 0.05
$\eta_2 = \eta_1 - \rho$	$\omega_2 = \eta_1 - \rho$
Stopping criterion is when there is no progress in the fitness value for successive iterations	Stopping criterion is when there is no progress in the fitness value for successive iterations

Description of COFS

Step 1:

Initialise the variables E and k where E represents the set of total available eggs and E = {e_1,e_2,e_3,…,e_n} where n is total number of available eggs. This is analogous to the total feature set F = {f_1,f_2,f_3,…,f_n} where f represents the feature and n is the total number of features present in the ND. k represents the size of the host nest HN where HN can hold k number of eggs. In the same way, SF represents the feature subset of F, and k represent the size of the SF (i.e. SF contains k number of features; totally k number of features can be selected at a time).

Step 2:

Find the objective value η_i of each egg e_i using the objective function $O(e_i)$. In feature selection terminology, find the IG φ_i for each f_i using the IG function IG (f_i, C) as in Equations (1)–(3) where IG represents the IG of feature f_i and C represents the class attribute of ND. The IG of each feature f_i is denoted as β_i. The expected information EX that is needed for the detection is determined using the class attributes C as in Equation (1) where c is the total number of distinct labels of the class attribute C and network flow data set ND.

$$\mathrm{EX}(\mathrm{ND}) = -\sum_{i=1}^{c} P_i \log_2 (P_i) \tag{1}$$

The required information for each feature IF (D) of the network flow data set ND is determined as depicted in Equation (2) where v denotes the total number of distinct values of the feature f.

$$I_f(\mathrm{ND}) = \sum_{j=1}^{v} \frac{|\mathrm{ND}_j|}{|\mathrm{ND}|} \times I(\mathrm{ND}_j) \tag{2}$$

$$\beta_i = \mathrm{IG}(f_i) = \mathrm{EX}(\mathrm{ND}) – I_f(\mathrm{ND}) \tag{3}$$

where β_i represents the IG of the feature f_i.

Step 3:

Select k number of best eggs based on their objective value η_i from E, from subset SE, and place into host nest HN. Equivalently, select k number of best features based on their IG value φ_i and form feature subset SF based on their IG value φ_i and SF⊂F.

Step 4:

Evaluate the fitness value of HN where β_i is the fitness value of ith iteration and β_i = Fv(HN) where Fv is the fitness function of the host nest HN. Similarly, find the detective accuracy ω_i of the ith iteration and ω_i = da (SF, C) where ω_i represents the detection accuracy, da (.) represents the detection accuracy function and C represents the class attributes. The detective accuracy is calculated as expressed in Equation (4) using na ïve Bayes classifier.

$$\omega\text{=detection accuracy (da)= } \frac{\mathrm{TP+TN}}{P+N} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{4}$$

where P represents the total number of instances belongs to the positive class label (not attack) present in the ND. N represents the number of instances belonging to the negative class label (attack) present in the ND. True positive (TP) represents the correctly classified or predicted positive labels. True negative (TN) represents the correctly classified or predicted negative labels. FP represents the incorrectly classified or predicted positive label. False negative (FN) represents the incorrectly classified or predicted negative label.

Step 5:

Check if the fitness value β_i satisfies the stopping criterion. This is analogous to checking whether the detective accuracy value ω_i satisfies the stopping criteria. Then rank and select the higher value of β_i over iterations and select its corresponding SE$_i$ as the set of selected significant eggs and terminate the loop. In the same way, rank and select the higher value of ω_i and select its corresponding SF$_i$ as the set of selected significant features and terminate the loop. If the fitness value does not satisfy the stopping criterion, update η_i of each egg e_i of SE with ρ (abandoned rate). In feature selection terminology, update φ_i of each feature f_i of SE with ρ. η_i of each egg e_i of SE as expressed in Equation (5) and go to Step 3. The stopping criterion is set as there is no progress in the fitness value for successive iterations.

$$\text{Update }(\omega_i)\text{= } \eta_i - \rho \tag{5}$$

C. Intrusion Detection Model Using COFS

The schematic diagram of development of intrusion detection model is shown in Fig. 3. Initially, the network flow data is collected and prepared as data set. Then, irrelevant and redundant features from the data set are removed using the COFS. The relevant features are then given to the na ïve Bayes classification algorithm to develop the intrusion-detection model. Subsequently, this model is deployed in the network or the host of the cloud depending on from where the data is collected to learn and build the model. The IDS is shown in Fig. 4. The anomaly or intruder packets are detected using this intrusion detection model and alert signal such as attack or normal is given based on the packet that is arrived in the network. Then, the intrusion prevention system takes preventive measures based on the alert message produced by the IDS.

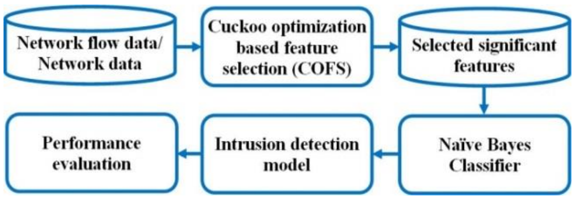
Network flow data/ Network data
Cuckoo optimization based feature selection (COFS)
Selected significant features
Naïve Bayes Classifier
Intrusion detection model
Performance evaluation

Fig.3. Schematic Diagram of Development of Intrusion Detection Model

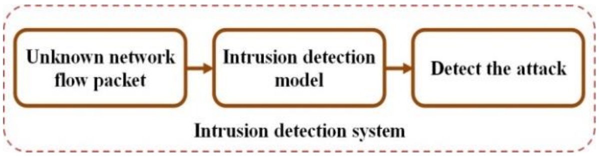
Unknown network flow packet
Intrusion detection model
Detect the attack
Intrusion detection system

Fig.4. Intrusion Detection System

IV. Implementation and Experimental Results

The experiment is conducted using NetBeans IDE 8.2 with Weka data-mining software [31] with the computer system specification of operating system: Windows 7 professional 64-bit, Processor: Intel(R) Core(TM)2 CPU and RAM: 4.00 GB. For the conduction of this experiment, the data set, namely NIMS 2 [32] is collected from the publically available data set repository. In this data set, the first 100 instances are taken from class

‘GTALK’, and first 100 instances are taken from class ‘PRIMUS’, and then the first 100 instances are from the class ‘ZFONE’ and all the 21 instances from class ‘ONLINE BANKING’ are taken to form the data set. Thus, the data set contains totally 321 instances, 22 features and 4 classes.

Table 2. Accuracy of NB Classifier with the Feature-selection Methods against the Number of Features Selected

Number of Selected Features	COFS	IGFS	CSFS	GRFS	ORFS
3	96.880	95.950	53.270	49.221	52.959
4	60.750	58.878	58.560	57.943	55.763
5	75.700	63.868	63.862	57.320	64.174
6	67.289	63.551	65.109	53.271	63.862
7	64.797	63.862	64.147	65.420	64.174
8	68.535	64.174	62.616	68.847	64.797
9	70.093	62.305	62.305	69.470	63.862
10	69.470	63.862	59.813	66.043	61.993
Average	71.689	67.056	61.210	60.941	61.448

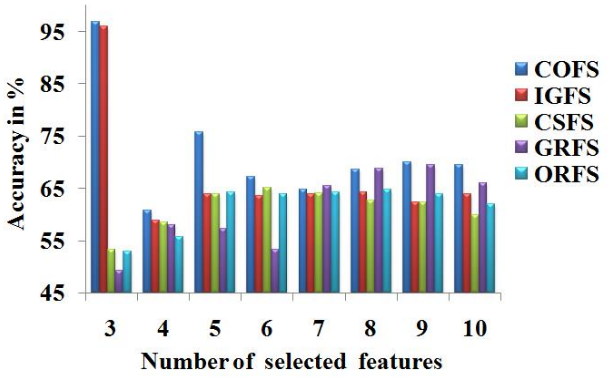

Fig.5. Accuracy of NB Classifier with the Feature-selection Methods against the Number of Features Selected

To compare the performance of the proposed system, the existing feature-selection algorithms, namely information gain-based feature selection (IGFS), chi-square feature selection (CSFS), information gain ratio-based feature selection (GRFS) and OneR feature selection (ORFS) [31] are used. Moreover, the naïve Bayes classification algorithm is employed to build the intrusion detection model and evaluate the performance of the feature-selection methods. The experiment is conducted as follows: Initially, the data set and the value of *k* (number of features to be selected) are given to each feature-selection method where *k* is the number of features to be selected. Then, *k* number of features selected by the feature-selection method along with the class attribute is given to the naïve Bayes classification algorithm [33], and the accuracy is calculated as tabulated in Table 2 to determine the performance of the feature-selection methods.

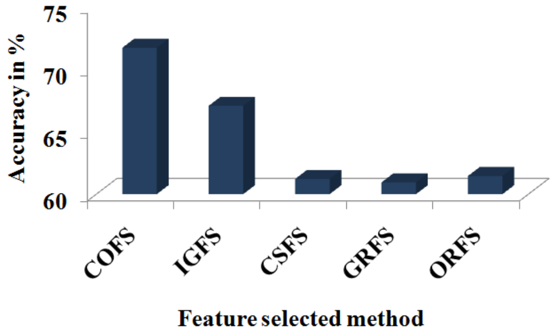

Fig.6. The Average NB Accuracy with the Feature-selection Method

From Table 2, Fig. 5 and Fig. 6, it is observed that the proposed method produces the better accuracy for the intrusion detection model compared with other methods.

V. Conclusion

This paper presented a COIDS. Moreover, to improve the accuracy of the IDS in the cloud environment, this paper proposed COFS. This proposed method is developed with the cuckoo optimisation-based searching technique to select the significant features from the ND with naïve Bayes classification algorithm. The performance of this proposed method is compared with different existing methods in terms of accuracy, and the performance of the proposed method is better than the other methods compared. This work can be extended with other optimisation techniques.

References

[1] Krutz RL, Vines RD. Cloud computing security architecture. Cloud security: a comprehensive guide to secure cloud computing. Indianapolis, IN: Wiley; 2010. pp. 179–80. Print.

[2] Tan Z, Nagar UT, He X, Nanda P, Liu RP, Wang S, Hu J. Enhancing big data security with collaborative intrusion detection. IEEE Transactions on Cloud Computing 2014; 1(3):27–33.

[3] Al-Jarrah OY, Alhussein O, Yoo PD, Muhaidat S, Taha K, Kim K. Data randomization and cluster-based partitioning for Botnet intrusion detection. IEEE Transactions on Cybernetics 2016;46(8):1796–806.

[4] Ambusaidi MA, He X, Nanda P, Tan Z. Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Transactions on Computers 2016;65(10):2986–98.

[5] El-Khatib K. Impact of feature reduction on the efficiency of wireless intrusion detection systems. IEEE Transactions on Parallel and Distributed Systems 2010; 21(8):1143–9.

[6] Mishra P, Pilli ES, Varadharajan V, Tupakula U. Intrusion detection techniques in cloud environment: a survey. Journal of Network and Computer Applications 2017; 77(2):18–47.

[7] Viegas E, Santin AO, França A, Jasinski R, Pedroni VA, Oliveira LS. Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems IEEE Transactions on Computers 2017; 66(1):163–77.

[8] Mistry K, Zhang L, Neoh SC, Lim CP, Fielding B. A micro-GA embedded PSO feature selection approach to

intelligent facial emotion recognition. IEEE Transactions on Cybernetics 2016; 47(6):1496–509.

[9] Lagrange A, Fauvel M, Grizonnet M. Large-scale feature selection with Gaussian mixture models for the classification of high dimensional remote sensing images. IEEE Transactions on Computational Imaging 2017; 3(2):230–42.

[10] Kaya GT, Kaya H, Bruzzone L. Feature selection based on high dimensional model representation for hyperspectral images. IEEE Transactions on Image Processing 2017; 26(6):2918–28.

[11] Wang Y, Wang J, Liao H, Chen H. Unsupervised feature selection based on Markov blanket and particle swarm optimization. Journal of Systems Engineering and Electronics 2017; 28(1):151–61.

[12] Ma L, Li M, Gao Y, Chen T, Ma X, Qu L. A novel wrapper approach for feature selection in object-based image classification using polygon-based cross-validation. IEEE Geoscience and Remote Sensing Letters 2017; 14(3):409–13.

[13] Huda S, Yearwood J, Jelinek HF, Hassan MM, Fortino G, Buckland M. A hybrid feature selection with ensemble classification for imbalanced healthcare data: a case study for brain tumor diagnosis. IEEE Access 2016; 4:9145–54.

[14] Nguyen TM, Wu QJ. Online feature selection based on fuzzy clustering and its applications. IEEE Transactions on Fuzzy Systems 2016; 24(6):1294–306.

[15] Abedinia O, Amjady N, Zareipour H. A new feature selection technique for load and price forecast of electrical power systems. IEEE Transactions on Power Systems 2017; 32(1):62–74.

[16] Yang Y, Xu HQ, Gao L, Yuan YB, McLaughlin K, Sezer S. Multidimensional intrusion detection system for IEC 61850-based SCADA networks. IEEE Transactions on Power Delivery 2017; 32(2):1068–78.

[17] Marchang N, Datta R, Das SK. A novel approach for efficient usage of intrusion detection system in mobile Ad Hoc networks. IEEE Transactions on Vehicular Technology 2017; 66(2):1684–95.

[18] Ha T, Yoon S, Risdianto AC, Kim J, Lim H. Suspicious flow forwarding for multiple intrusion detection systems on software-defined networks. IEEE Network 2016; 30(6):22–7.

[19] Zhou C, Huang S, Xiong N, Yang SH, Li H, Qin Y, Li X. Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation. IEEE Transactions on Systems, Man, and Cybernetics: Systems 2015; 45(10):1345–60.

[20] Lo CH, Ansari N. Consumer: a novel hybrid intrusion detection system for distribution networks in smart grid. IEEE Transactions on Emerging Topics in Computing 2013; 1(1):33–44.

[21] Durbha SS, King RL, Younan NH, Wrapper-based feature subset selection for rapid image information mining. IEEE Geoscience and Remote Sensing Letters 2010; 7(1):43–7.

[22] Singh DAAG, Leavline EJ, Priyanka R, Priya PP. Dimensionality reduction using genetic algorithm for improving accuracy in medical diagnosis. International Journal of Intelligent Systems and Applications 2016; 8(1):67–73.

[23] Singh DAAG, Leavline EJ, Valliyappan K, Srinivasan M. Enhancing the performance of classifier using particle swarm optimization (PSO)-based dimensionality reduction. International Journal of Energy, Information and Communications 2015; 6:19–26.

[24] Singh DAAG, Leavline EJ, Priyanka V, Swathi V. Agriculture classification system using differential evolution algorithm. International Advanced Research Journal in Science, Engineering and Technology 2016; 3:24–8.

[25] Singh DAAG, Leavline EJ, Nithya T, Nivetha S. Artificial immune system based organizational data prediction. International Journal of Engineering Science 2016; 6:4633–7.

[26] Singh DAAG, Surenther P, Leavline EJ. Ant colony optimization based attribute reduction for disease diagnostic system. International Journal of Applied Engineering Research 2015; 10(55):156–565.

[27] Liao Q, Zhou S, Shi H, Shi W. Parameter estimation of nonlinear systems by dynamic cuckoo search. Neural Computation 2017; 29(4):1103–23.

[28] Jiang M, Luo J, Jiang D, Xiong J, Song H, Shen J. A cuckoo search-support vector machine model for predicting dynamic measurement errors of sensors. IEEE Access 2016; 4:5030–7.

[29] Cheung NJ, Ding XM, Shen HB. A nonhomogeneous cuckoo search algorithm based on quantum mechanism for real parameter optimization. IEEE Transactions on Cybernetics 2017; 47(2):391–402.

[30] Kulshestha G, Agarwal A, Mittal A, Sahoo A. Hybrid cuckoo search algorithm for simultaneous feature and classifier selection. In: IEEE International Conference on Cognitive Computing and Information Processing (CCIP); March 2015. pp. 1–6.

[31] Frank E, Hall MA, Witten IH. The WEKA workbench. In: Kaufmann M, editor. Online appendix for data mining: practical machine learning tools and techniques. 4th ed. 2016.

[32] Alshammari R, Zincir-Heywood AN. An investigation on the identification of VoIP traffic: case study on Gtalk and Skype. In: 6th International Conference on Network and Services Management CNSM 2010). Niagara Falls, Canada, October 2010. pp. 25–9. URL: https://web.cs.dal.ca/~riyad/Site/Download.html.

[33] Singh G, Antony DA, Leavline EJ. Data mining in network security-techniques and tools: a research perspective. Journal of Theoretical and Applied Information Technology 2013; 57(2):269–78.

Authors' Profiles

Dr. D. Asir Antony Gnana Singh received the Bachelor of Engineering in Computer Science and Engineering, Master of Engineering in Computer Science and Engineering, Master of Business Administrator in Human Resource Management, and Ph. D in Information and Communication Engineering degrees from Anna University, India. He is currently working as a teaching fellow in the Department of Computer Science and Engineering, Anna University, BIT-Campus, Tiruchirappalli, India. His research interests include data mining, wireless networks, parallel computing, mobile computing, computer networks, image processing, software engineering, soft computing, cloud computing, big data analytics, teaching learning process and engineering education, human resource management.

R. Priyadharshini is doing her post-graduation at Department of Computer Science and Engineering, Anna University, BIT-Campus, Tiruchirappalli, India. Her field of interests includes cloud computing, information security, computer networks, machine learning, data mining, big data analytics, and computer vision.

Dr. E. Jebamalar Leavline received the Ph.D, M. Eng. and B. Eng. degrees from Anna University, India, and received the MBA degree from Alagappa University, India. She is currently working as an assistant professor in the Department of Electronics and Communication Engineering, Anna University, BIT-Campus, Tiruchirappalli, India. Her research interests include image processing, signal processing, VLSI design, data mining, teaching learning process and engineering education.

How to cite this paper: D. Asir Antony Gnana Singh, R. Priyadharshini, E. Jebamalar Leavline,"Cuckoo Optimisation based Intrusion Detection System for Cloud Computing", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.11, pp.42-49, 2018.DOI: 10.5815/ijcnis.2018.11.05