S. Immaculate Shyla* and S.S. Sujatha

Cloud Security: LKM and Optimal Fuzzy System for Intrusion Detection in Cloud Environment

https://doi.org/10.1515/jisys-2018-0479
Received December 10, 2018; previously published online November 15, 2019.

Abstract: In cloud security, intrusion detection system (IDS) is one of the challenging research areas. In a cloud environment, security incidents such as denial of service, scanning, malware code injection, virus, worm, and password cracking are getting usual. These attacks surely affect the company and may develop a financial loss if not distinguished in time. Therefore, securing the cloud from these types of attack is very much needed. To discover the problem, this paper suggests a novel IDS established on a combination of a leader-based k-means clustering (LKM), optimal fuzzy logic system. Here, at first, the input dataset is grouped into clusters with the use of LKM. Then, cluster data are afforded to the fuzzy logic system (FLS). Here, normal and abnormal data are inquired by the FLS, while FLS training is done by the grey wolf optimization algorithm through maximizing the rules. The clouds simulator and NSL-Knowledge Discovery and DataBase (KDD) Cup 99 dataset are applied to inquire about the suggested method. Precision, recall, and F-measure are conceived as evaluation criteria. The obtained results have denoted the superiority of the suggested method in comparison with other methods.

Keywords: Intrusion detection system, cloud computing, cloud security, grey wolf optimization, leader-based k-means clustering, fuzzy logic system, KDD Cup 99.

1 Introduction

Nowadays, cloud computing [23] renders data storage and computing services through the Internet. Cloud computing has speed, scalability, and elasticity, etc. Cloud computing is a general term for anything that admits delivering hosted services over the Internet and managing the data, by the cloud service provider (CSP). At remote locations, the cloud services permit businesses and people to use software and hardware infrastructure that is led by third parties. An increasing number of cloud users raises privacy and security concerns. Data protection becomes a major issue as the user's data are handled by a third party [15]. The number of attacks on computer networks has grown extensively, various new hacking tools and intrusive methods have emerged on a widespread basis. Within a network, using an intrusion detection system (IDS) is one way of handling suspicious activities [22]. An IDS monitors the activities of an afforded environment and decides whether these activities are malicious (intrusive) or legitimate (normal), demonstrated on system integrity, confidentiality, and the availability of information origins [9].

The IDSs may be changed to perform misuse detection or anomaly detection in general [4]. All known abnormal behavior is evaluated, and the system is trained to identify it in misuse detection. It works by equating arriving packet with features of known attack behavior. If any new, not predefined attack arrives, the system would distinguish it as a normal packet, inducing high false negative rate (FNR) [10]. To avoid very high

***Corresponding author: S. Immaculate Shyla,** Department of Computer Science, S.T. Hindu College, Nagercoil, India; and Manonmaniam Sundaranar University, Tirunelveli, India, e-mail: immaculateshylas1918@gmail.com. https://orcid.org/0000-0001-5390-3981
S.S. Sujatha: Department of Computer Science and Applications, S.T. Hindu College, Nagercoil, India; and Manonmaniam Sundaranar University, Tirunelveli, India

FNR, misuse-based IDS must be retrained very often, sometimes inducing delays in the network [21]. A number of data mining techniques have been introduced to resolve the limitations of the above methods [18]. In the data, an artificial neural network (ANN) is an efficient algorithm to inquire about the intrusion present. However, ANN also has some drawback such as lower detection precision, especially for low-frequency attacks, e.g. Remote to Local (R2L), User to Root (U2R), and weaker detection stability [24].

To provide a better detection technique for inquiring about the intrusion from the dataset by resolving the issues that currently exist in the literary works is a major aim of this research. Hence, for the IDS, we have intended to suggest a novel detection method. Our suggested method contained three stages, namely, clustering, training, and testing. Primarily, we separate the dataset into two subsets such as training and testing. Then, the training dataset is extracted from the given input database. Then, to reduce the complexity, the training data are clustered using leader-based k-means clustering (LKM) algorithm. Then, we train the subset applying the optimal fuzzy logic system (OFLS). In this FLS, the optimal rules are selected using grey wolf optimization (GWO), which will be used to reduce the time complexity and increase the detection accuracy. Finally, based on the fuzzy score, the data are classified as normal or abnormal. The rest of the paper is organized as follows: a brief review of researches associated with the proposed technique is introduced in Section 2. In Section 3, the authors explain the background of the research and suggested IDS. The detailed experimental results and discussions are explained in Section 4. The conclusion is summed up in Section 5.

2 Related Work

Researchers are more interested in intrusion detection since it is usually maintaining security over the network in the current days. Here, they referred to some of the intrusion detection techniques. Bahram and Nima [8] have implemented IDS and demonstrated its combination of multilayer perceptron (MLP) network, artificial bee colony (ABC) and fuzzy clustering algorithms. Moreover, a honeypot based strategy for intrusion detection/prevention systems has been suggested by Baykara and Das [2]. The developed honey pot server application was combined with IDSs to tested data in real time and to control effectively. Moreover, by equating the advantages of low- and high-interaction honeypots, a superior hybrid honeypot system was performed. Mehrnaz et al. [14] have implemented the reliable hybrid method for an anomaly network-based IDS using ABC and AdaBoost algorithms in order to gain a high detection rate with the low false positive rate.

Similarly, the Collaborative Study of Intrusion Detection and Prevention Techniques in Cloud Computing has been explained in Shadab et al. [1]. Hypervisor-based and distributed IDSs have shown promising security features in a cloud computing environment in comparison with traditional identity provider techniques. Partha et al. [6] have presented intrusion detection in the cloud using hybridization of the cuckoo search algorithm and particle swarm optimization (PSO) algorithm. Moreover, Fang et al. [5] have explained anomaly detection in an ad hoc network using deep learning algorithm. Here, they utilize a plug and play device to detect denial of service (DoS) and privacy attacks. Sohal et al. [20] have introduced a digital security system. Their structure has been completely shown to distinguish the malicious edge gadgets in the circulated fog computing condition. Similarly, Girma et al. [7] have exhibited a propelled machine learning way to deal with identifying the DoS assaults on cloud computing with entropy utilizing clustering innovation. They were proceeding with this research to execute those extremely compelling distributed DoS hybrid detection system. Kozik et al. [11] have presented a distributed extreme learning machine technology based attack detection approach that uses cluster resources. Moreover, Bhushan and Gupta [3] have examined different basic features of software defined network that makes it an appropriate systems administration innovation for cloud computing. In addition, they speak to the stream table space of a switch by utilizing a lining hypothesis based numerical model.

Zeenat et al. [12] have explained a principal component analysis (PCA) and neural network (NN) based intrusion detection. This work takes maximum time to find out the intrusion data. In [17], Mehdi and Mohammad have explained a NN based on a different attack detection. Here, four types of attacks are identified

with the help of a NN. Moreover, Manickam et al. [13], have explained a probabilistic fuzzy c-means clustering (PFCM) and recurrent neural network (RNN) based IDS. Here, PFCM classifier was utilized for clustering process, and RNN is used for classification. Here, also, four types of attacks are identified.

3 Proposed Model for the IDS

Cloud computing manages parts of assets and computing offices through the Internet. Cloud frameworks pull in numerous clients with its attractive features. Notwithstanding them, cloud frameworks may encounter serious security issues. In order to improve the security of the cloud system, here we have intended to propose an efficient IDS. The main objective of the proposed methodology is to design cloud IDS for achieving cloud security. To achieve the security of the system, in this paper we develop an algorithm based on LKM algorithm and OFLS. Here, with the proposed FLS, the rules are optimally selected with the help of GWO algorithm. The overall process of the proposed technique is shown in Figure 1.

In the proposed technique, at first, the input data are preprocessed. After preprocessing, we cluster the preprocessed data using kernel LKM. After that, each clustered data are given to the OFLS to detect the data as normal or intruded data. Then, finally, the normal data are stored on the cloud. The overall process is split

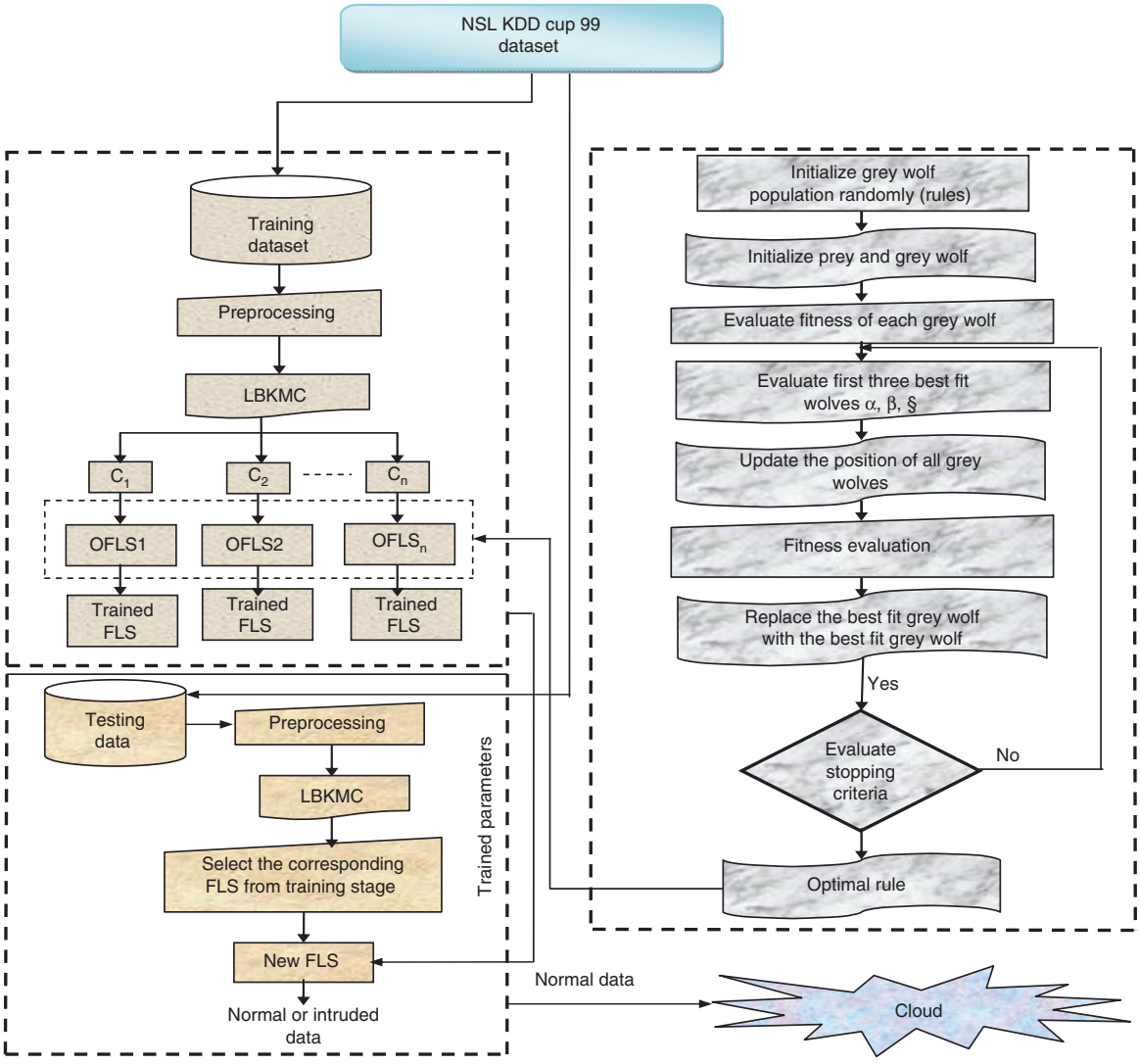

Figure 1: Overall Diagram of a Proposed Intrusion Detection System.

into two stages, namely, training and testing. The dataset utilized in this proposed method is NSL-Knowledge Discovery and DataBase (KDD) CUP 99 dataset. The step by step process is described in a further section. The proposed method has three main processes, namely,

- Preprocessing
- Clustering
- Intrusion detection

3.1 Preprocessing

Consider the NSL-KDD dataset which consists of $\boldsymbol{n}$ number of records and 41 features in which the data may be incomplete, noisy, or duplicate. Therefore, before starting the IDS process, we have to preprocess the data. The preprocessed outputs provide optimal data to the IDS, and this will increase the detection accuracy. The steps involved in preprocessing are given below:

- The symbolic attributes in the dataset are converted into the numeric value.
- Then, the numeric attributes are normalized. Let X_{ij} represent the jth column attribute value in the ith row of the dataset and M_i represent the mean value of the jth column attribute. The normalization is done using equation (1).

$$X_{ij} = \frac{X_{ij} - M_i}{(\text{Max} - \text{Min}) \text{ value of feature } j} \tag{1}$$

After the normalization process, the data are given to the clustering process.

3.2 LKM Based Clustering Module

The aim of the leader-based clustering module is to partition an afforded set of data into clusters and also this algorithm mainly used to speed up the k-means clustering algorithm. Consider the dataset S which consists of $\boldsymbol{n}$ number of data and $\boldsymbol{m}$ number of attributes. To handle a large number of data is hard for processing. Therefore, at first, we cluster the dataset into $\boldsymbol{a}$ number of clusters in order to decrease the size of the training subset and complexity of the IDS. For that, the LKM algorithm is applied to the clustering process.

To partition the dataset into $\boldsymbol{a}$ number of clusters, the leader clustering method takes the size of each cluster, called the threshold T, as an input parameter [19]. Clusters are mentioned by a pattern as a leader, and other patterns in the cluster are mentioned as followers. The set of leaders A is maintained initially empty and is incrementally built. If there is, leader $a \in A$ such as distance between u and A is less than or equal to T for each pattern in the dataset S, then the pattern is assigned to the cluster represented by a. In this case, we call patterns as a follower of the leader and the leader is a follower of itself. The first user, which is at a distance less than or equal to T, is chosen as a follower of the leader. The pattern u becomes a new leader if there is no such leader and is added to A. The set of leaders A is provided as output by the algorithm. Modifications used in this proposed method are as follows:

- The clusters are not found in input space, and it can be found only in kernel space.
- According to the pattern in the input space, each cluster is represented by its leader.
- All the patterns in each cluster can be retrieved easily when the datasets are re-indexed according to these clusters.
- The principle behind the proposed kernel based leaders clustering method is its linear time complexity. Based on the size of the input, the running time increases linearly, and its working principle is as follows.

For a given threshold T, a set of leaders A and the number of followers of each leader A is maintained by the kernel based leaders clustering method, which is count a. A is initially empty and is incrementally built. For each pattern u in the dataset S, if there is a leader $a \in A$, such that the distance between $\varphi(u)$ and $\varphi(a)$ is less than or equal to T, then u is assigned to the cluster that is represented by a count (a), and the value is incremented by 1. Otherwise u becomes a new leader and is added to A, and count (a) becomes 1. The output

of the algorithm is the set of leaders A, the number of followers of each leader, i.e. count (a) and the set of followers of each leader a, i.e. followers (a). This output is denoted by $A*$. The proposed kernel based leaders clustering method is given in Table 1.

Stage 1: First, the kernel based leaders clustering method is used to find $A*$.

Stage 2: Later, to derive a partition of the set of leaders ρ_A, in the set of leaders A which is taken from $A*$, applied again the kernel k-means clustering method. In all iterations, each leader a_i is assigned to the cluster C_r such that $\|\varphi(a_i) - m_r^2\|$ is minimized. Assume that the patterns in the cluster are very close to the leader where it exists. Hence, $\|\varphi(a_i) - m_r^2\|$ is computed as follows.

$$\|\varphi(a_i) - m_r\|^2 = \left\| \varphi(a_i) - \sum \frac{\varphi(a_r)}{\left(\sum_{a_r \in C_r} \text{count}(a_r)\right)} \right\|^2 \tag{2}$$

$$= \varphi(a_i) \cdot \varphi(a_i) - J(a_i, C_r) + L(C_r) \tag{3}$$

where

$$J(a_i, C_r) = \frac{2}{\left(\sum_{a_r \in (C_r)} \text{count}(a_r)\right)} \sum_{a_i \in C_r} \{\text{count}(a_r) K(a_i, a_r)\}, \tag{4}$$

$$L(C_r) = \frac{1}{\left(\sum_{a_r \in (C_r)} \text{count}(a_r)\right)^2} \{B_1 + B_2\}. \tag{5}$$

where

$$B_1 = \sum_{a_r \in C_r} \left\{\text{count}(a_r)^2 K(a_r, a_r)\right\}, \tag{6}$$

$$B_2 = \sum_{a_r \in C_r} \sum_{a_s \in C_r} \{\text{count}(a_s) K(a_r, a_s)\}, \quad \text{for } a \neq s \tag{7}$$

Finally, each leader is replaced by all of its followers to get a partition of the entire dataset at the end of the iterative process, and it is denoted by ρ_S^*. The proposed method is explained below.

3.3 IDS Using OFLS Classifier

After the clustering process, each obtained cluster is given to the OFLS. The number of clusters and the OFLS are identical. In this, the FLS rules are optimally selected using a bio-inspired algorithm, namely, GWO.

Table 1: Algorithm for Kernel-Based Leaders Clustering Method.

Algorithm 1: Kernel-based leaders clustering method (S, T)

$A = 0$
for each $u \in S$ **do**
Find a leader $a \in A$ such that $\|\varphi(a) - \varphi(u)\| \leq T/*$ where
$\|\varphi(a) - \varphi(u)\|$ can be computed using the equation
if there is no such A or when $A = 0$, **then**
$A = A \cup \{u\}$;
count(u) = 1;
followers(u) = $\{u\}$;
else
count(a) = count(a) + 1;
followers(a) = followers(a) $\cup \{x\}$
end if
end for
Output:
$A* = \{<a, \text{count}(a), \text{followers}(a) > a \text{ is a leader}\}$

Table 2: Proposed Prototype Based Hybrid Kernel k-Means.

Prototype based hybrid kernel k-means (D, k, $\varepsilon^{(0)}$, T)
Step 1: $A*$ is generated by using the kernel-based leaders clustering method that is given in the algorithm.
Step 2: Using the given initial seed points $\varepsilon^{(0)}$, compute the initial partition $\rho_A^{(0)}$ of the leader set A.
Step 3: Apply kernel k-means clustering method $\left(\boldsymbol{A}, \boldsymbol{k}, \boldsymbol{\rho}_A^{(0)}\right)$ and find the nearest cluster for a leader. Let ρ_A be the output.
Step 4: To get the partition for the entire dataset, say ρ_S^*, replace each leader $a \in \rho_A$, by its cluster.
Step 5: Output is $\boldsymbol{\rho}_D^*$.

A fuzzy set can address and handle uncertain data successfully. Table 2 shows proposed protiotype based hybrid kernel k-means. The database D is divided into two sets, namely, training (D_{TR}) and testing (D_{TE}). The training data are used to generate the FLS system. The intrusion detection accuracy of the proposed system is evaluated with the help of testing dataset. I have used the 494,000 records. I have taken the records for 80% (395,200) of data for training and 20% (98,800) for testing.

3.3.1 Training Process

After the clustering process, each output of the clusters is trained applying N number of fuzzy logic classifier (FLC). Here, the number of clusters and FLC are same. A FLS is distinctive in that it is competent to handle the numerical data and linguistic knowledge. In this FLS, rules are optimally selected with the help of GWO algorithm [16]. The training process is given in Figure 2.

3.3.1.1 Optimal Rule Generations Using GWO

GWO algorithm is applied to choose the best rule for prediction. In this section, we have N number of clusters, and each cluster has M number of data. Each data has S number of attributes. We utilize the NSL-KDD Cup 99 dataset. Here, each data has 41 features in this paper. As established on the attribute range we generate the rule. Then, we optimally choose the rule using the GWO algorithm. Primarily, the database D is divided into two sets, a training dataset (D_{TR}) and analyzing dataset (D_{TE}). The training dataset is applied to generate the fuzzy rules and the aligning of the fuzzy system. With the help of the testing dataset, the prediction accuracy of the suggested system is estimated. The detailed process of generating the rule generation applying GWO algorithm is explained by applying the following steps.

- **Discretization**
  At first, we consider the training dataset D_{TR}, which comprise of S number of attributes and N number of data. Here, a number of features are given to a discretization function in order to transfer the input data into a discretized one. The main property of discretization is to change the data value into the specific interval, which means the range of data value is changed into a specific interval. The discretization process is explained below:

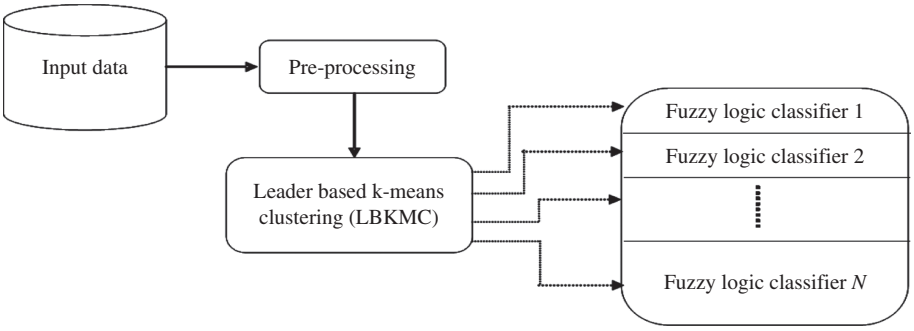

Figure 2: Cluster from the LBKMC is Trained to Apply N Number Fuzzy Logic Classifier.

Step 1: Consider the training dataset D_{TR}, and then we take the attributes in columns. Each column comprises N number of data, and each row comprises n features; $0 < n \leq 41$.

Step 2: Then, we calculate the median (M_{med}) of each column j.

Step 3: After that, we change the data value into a particular range using the M_{med} value. Here, the particular data value is divided into M_{med} value. Consequently, separate all the feature values, in particular the median value M_{med}. For example, if the median value $M_{\text{med}} = 100$ and feature value $P_M^{(1)} = 50$, it means feature value as 0.5 if it uses the following equation:

$$P_M = \frac{M_{\text{med}}}{P_M^{(1)}} \tag{8}$$

Applying equation (7), we can alter all the feature values in the specified interval. Now we found the new feature values, which vary from 0 to 1. Then, every value that comes within the range is aligned with the interval value so that the input data are transformed into the discretized data. Consequently, the training dataset D_{TR} is concerted to the discretized format D_D where the entire data element D_D comprises only the L, M, and H.

$$F_M^i = \begin{cases} L0 < P_M < 0.5 \\ M0.5 \leq P_M < 0.75 \\ HP_M \geq 0.75 \end{cases} \tag{9}$$

where
$P_M^i \rightarrow i$th feature of the dataset
$L \rightarrow$ low value of the feature
$M \rightarrow$ medium value of the feature
$H \rightarrow$ high value of the feature
After discretization function, the training dataset D_{TR} is changed into the discretized format D_D where all the data elements $D_D(i, j)$ contain only the L, M, or H if $k = 3$

- **Logical rule generation**
  As established on the discretized format I^D, we generate the rule; here the rule should have two decisions such as N or AT. The sample rules are afforded in Table 3.
  As established on the rule, we rearrange the dataset D_D into two groups, each group having only one kind of data.
- **Solution representation**
  To optimize the rules, GWO algorithm primarily creates an arbitrary population of the solution. Solution creation is a crucial step of an optimization algorithm that helps to identify the optimal solution quickly. Each solution consists of one rule, and that rule is filled with L, H, and M values. The sample rule is afforded in Table 3.

Table 3: Sample Rules are Fuzzy Logic.

Rules	$P_M^{(1)}$	$P_M^{(2)}$	$P_M^{(3)}$	$P_M^{(4)}$	.....	$P_M^{(41)}$	Decision
1	M	H	L	L	.....	L	*N*
2	L	M	H	M	.....	H	*N*
3	H	L	M	H	.....	M	*AN*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	H	H	L	M	.....	M	*AN*

Table 4: Solution Encoding.

Rules	Randomly generated rules
R_1	IF (P_1 is L) and (P_2 is H) and (P_3 is L) and (P_4 is M) and . . . . (P_{41} is H) THEN Decision = Normal (N)
R_2:	IF (P_1 is H) and (P_2 is L) and (P_3 is M) and (P_4 is H) and . . . . (P_{41} is L) THEN Decision = Abnormal (AN)
⋮	⋮
R_k	IF (P_1 is M) and (P_2 is H) and (P_3 is L) and (P_4 is L) and . . . . (P_{41} is H) THEN Decision = Normal (N)

- **Fitness computation**
  The selection of fitness is an important aspect in the GWO algorithm Table 4 shows solution encoding. It is utilized to assess the aptitude (goodness) of candidate solutions. Here, the precision value is the major criterion used to design a fitness function. The fitness function is afforded in equation (10).

$$\text{Fitness} = \max(P) \tag{10}$$

$$P = \frac{TP}{TP + FP} \tag{11}$$

  where *TP* refers to the true positive and *FP* indicates the false positive value.
- **Assigning the best solution**
  After the fitness calculation, we have to assign the first, second, and third best values as S_α, S_β, and S_γ, respectively.
- **Encircling prey**
  The hunting is guided by α, β, and δ, and ω trails these three candidates. In order for the pack to hunt prey, the pack is first encircling it.

$$X(t+1) = X(t) - \vec{A}.\vec{K} \tag{12}$$

$$\vec{K} = \left|\vec{C}.X(t+1) - X(t)\right| \tag{13}$$

$$\vec{A} = 2\vec{a}r_1 - \vec{a} \quad \text{and} \quad \vec{C} = 2r_2 \tag{14}$$

  where t is iteration number, $X(t)$ is prey position, A and C are coefficient vectors, $\vec{a}$ is linearly decreased from 2 to 0, and r_1, $r_2 \rightarrow$ random vector [0, 1].
- **Hunting**
  We undertake that the α (best candidate solution), β, and δ have enhanced information about the potential site of the prey to replicate scientifically the hunting performance of the grey wolves. As a solution, we store the first three best results reached so far and need the other search agents (as well as the omegas) to study their positions permitting to the position of the best search agent. For recurrence, the novel solution $X(t+1)$ is assessed with the help of the formulae revealed as follows.

$$\vec{K}^\alpha = \left|\vec{C}_1 \cdot F_\alpha - F\right|, \quad \vec{K}^\beta = \left|\vec{C}_2 \cdot F_\beta - F\right|, \quad \vec{K}^\delta = \left|\vec{C}_3 \cdot F_\delta - F\right| \tag{15}$$

$$F_1 = F_\alpha - \vec{A}_1 \cdot (\vec{K}^\alpha), F_2 = F_\beta - \vec{A}_2 \cdot (\vec{K}^\beta), F_3 = F_\delta - \vec{A}_3 \cdot (\vec{K}^\delta) \tag{16}$$

$$F(t+1) = \frac{F_1 + F_2 + F_3}{3} \tag{17}$$

  It can be perceived that the concluding position would be in a random place including a circle that is distinct using the points of α, β, and δ in the search space. In added arguments, α, β, and δ evaluate the position of the prey, and other wolves inform their positions arbitrarily near the prey.

– **Attacking prey (exploitation) and search for prey (exploration)**
Exploration and exploitation are failsafe with the help of the adaptive values of b and B. The adaptive values of parameters b and B let GWO to effortlessly transition among exploration and exploitation. With declining A, half of the repetitions are dedicated to exploration ($|B| \geq 1$), and the rest half are devoted to exploitation ($|A| < 1$). The GWO has only two chief parameters to be accustomed (b and C), though we have retained the GWO algorithm as humble as likely with the smallest operators to be accustomed. The procedure will be sustained until the maximum number of iteration is obtained. Lastly, optimal solutions are designated on the basis of the fitness value.
– **Termination criteria**
Stop if the maximum number of generations is achieved. The best rule is selected and given to the FLS for further processing, after a suitable training process, we can decide if the data under test is normal or abnormal.

3.3.1.2 Fuzzy System Design

After the optimal rule generation, we are aligning the fuzzy system. When we are designing the fuzzy system, the fuzzy membership function (MF) definition and fuzzy rule base are the two important steps.

– **Membership function**
The formula used to compute the membership values is depicted as below. A MF is a curve that evaluates how each point in the input space is mapped to a membership value (or degree of membership) among 0 and 1. Moreover, the MF is aligned by choosing the proper MF. Here, we have selected the triangular MF to change the input data into the fuzzified value. The triangular MF comprises three vertices a, b, and c of $f(x)$ in a fuzzy set A (a: lower boundary and c: upper boundary where membership degree is 0, b: the center where membership degree is (1). One of the key issues in all fuzzy sets is how to estimate fuzzy MFs.
  – The MF fully evaluates the fuzzy set.
  – A MF renders and the measure of the degree of similarity of an element to a fuzzy set.
  – MFs can take any form, but there are some usual examples that appear in real applications.
  The formula used to compute the membership values is described below,

$$f(x) = \begin{cases} 0 & \text{if } x \leq a \\ \dfrac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \dfrac{c-x}{c-b} & \text{if } b \leq x \leq c \\ 0 & \text{if } x \geq c \end{cases} \tag{18}$$

– **Rule-based fuzzy score computation**
Using GWO optimization algorithm, we already generated the fuzzy rule set (refer to Section 4.1 Dataset Description). These rules are afforded to the fuzzy logic. The rule base contains a set of fuzzy rules in the form of low, high, and medium distance values.

3.3.2 Testing Module

After the training process, we test the incoming data. In the testing process, the cloud user uploads the data to the CSP. In this stage, the CSP checks whether incoming data are normal or intruded because the CSP is not aware of incoming data. In training, at first, the incoming data are preprocessed. Then, the preprocessed data are given to the clustering process. After the clustering process, the data are given to the corresponding cluster based FLS. The trained FLS structure tests the data. Finally, we obtained the score value. Based on the score value, we check whether the given data are intruded or not. In this, based on the score value, we fix one threshold T_h. If the obtained score value is above the threshold T_h, it means the data are intruded; otherwise

Table 5: Overall Algorithm.

Input: NSL-KDD cup dataset
Parameters of LBKMC, neural network and GWO
Output: classified normal data and abnormal data
1. Select uploading data
2. Apply preprocessing process in selected data (refer to Section 4.1)
3. Call Section 4.2 to the clustering process
4. Apply optimal fuzzy system to each clustered output
5. Call discretization process
6. Call logical rule generation process
7. Initialize random rule for GWO
8. Call fitness function
9. Call GWO operators
10. Select optimal rule
11. Design a fuzzy system based on the optimal rule
12. Detect normal or intruded data using fuzzy score
13. Output (normal or intrudes)
14. Store normal data on the cloud

the data are normal. Thus, the obtained score value satisfies the condition which is given in equation (19), and the overall algorithm is given in Table 5.

$$\text{result} = \begin{cases} T_h \geq \text{score}; & \text{data are normal} \\ T_h < \text{score}; & \text{data are intruded} \end{cases} \tag{19}$$

4 Results and Discussion

This section affords the detailed view of the result that is found by our proposed intrusion detection in cloud applying LKM and optimal FLS, which is performed in the working platform of JAVA with Cloud Sim tools and a series of experiments performed on a PC with Windows 7 Operating system at 2 GHz dual-core PC machine with 4 GB main memory running a 64-bit version of Windows 2007. To estimate the performance of the suggested LKM+OFLS based intrusion detection method, a series of experiments on the NSL-KDD CUP1999 dataset were conducted.

4.1 Dataset Description

The NSL-KDD dataset is a refined version of its predecessor KDD"99 dataset, and this dataset is widely applied for the IDS. This dataset contains five million records, and each record consists of 41 features. The attack classes present in the NSL-KDD dataset are grouped into four classes, namely, Probe attacks, U2R attacks, R2L attacks, and DoS. This dataset has a binary class attribute. Also, it has a reasonable number of training and test instances which makes it practical to run the experiments on.

4.2 Evaluation Metrics

The evaluation of the suggested IDS is carried out applying the following metrics as proposed by equations given below:

Precision: Precision is the ratio of the number of normal data inquired to the total number of normal and abnormal data detected, which is afforded in equation (20).

$$P = \frac{TP}{TP + FP} \tag{20}$$

Recall: Recall is the ratio of the number of normal data inquired to the total number of data present in the dataset, which is afforded in equation (21).

$$R = \frac{TP}{TP + FN} \tag{21}$$

F-measure: F-measure is determined as the harmonic mean of precision and recalls metrics, which is afforded in equation (22).

$$F = \frac{2PR}{P + R} \tag{22}$$

where
TP → true positive, FP → false positive, FN → false negative.

4.3 Simulation Results

The simulation results obtained from the proposed methodology is given in this section. The simulation is done on the working platform of JAVA with Cloud Sim tools. The proposed methodology test bed is given in Figure 3. Moreover, Figures 4–8 show the simulation results obtained from the proposed IDS.

The proposed IDS can be used in real-time applications. For real-time analysis, due to the lack of storage place and security, *n* numbers of users want to upload their data on the cloud. During the process of data uploading, an intrusion detector in the cloud detects or classifies normal or intruded data using the proposed algorithm (LKM+OFLS). At the end of verification or detection, the normal data are stored on the cloud, and intruded data are neglected. Hence, this process will increase the storage of the cloud. The proposed text bed is given in Figure 3.

4.4 Performance Analysis

The aim of the suggested methodology is to inquire whether the data are normal or intruded applying a combination of clustering and classifier techniques. Here, at first, the data are pre-processed to make it fit for further processing. Then, the preprocessed data are afforded to the clustering process. We have used a LKM algorithm for subset of the data into *n* numbers. Then, each subset is afforded to a separate fuzzy logic system. Finally, established on fuzzy logic score value, we identified the afforded data as normal or intruded data.

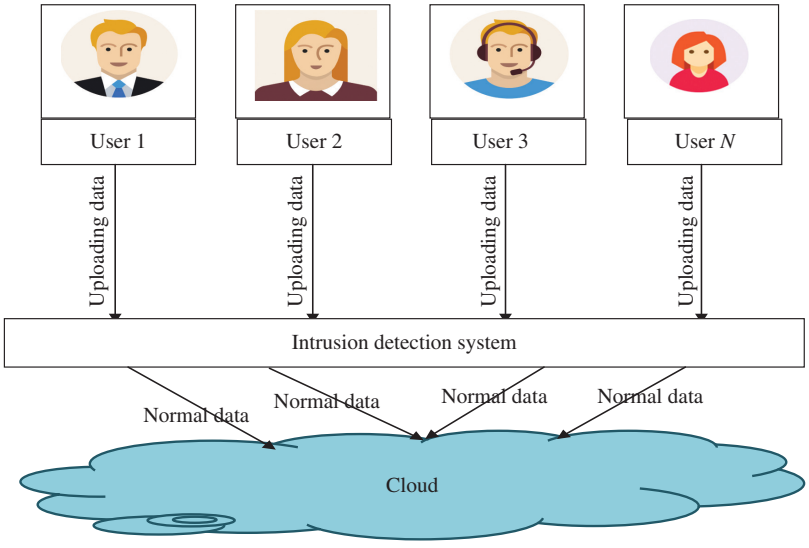

Figure 3: Testbed of the Proposed Approach.

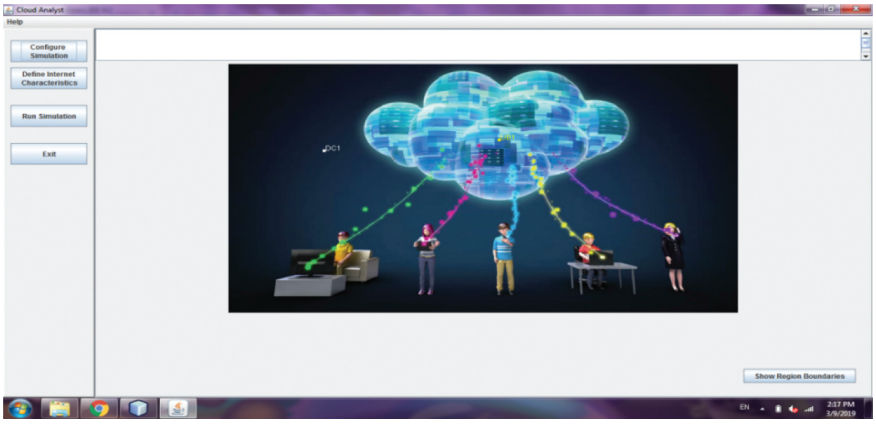

Figure 4: Cloud Simulation Window.

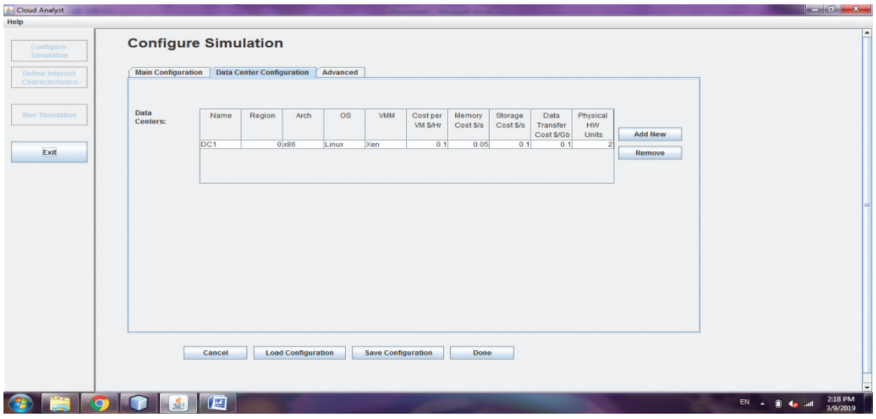

Figure 5: New Data Center Creation Window.

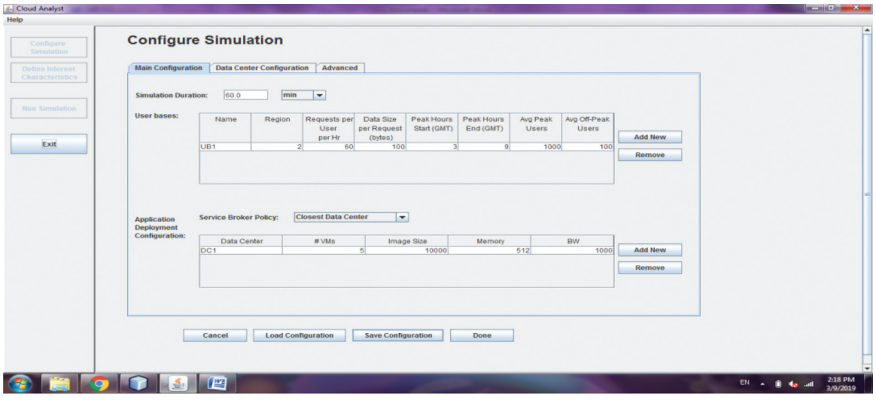

Figure 6: Data Center Adding Window.

We analyze the performance applying precision, recall, and F-measure by varying cluster size and data size by this paper. The performance of the suggested methodology is afforded in Figures 9–11.

Figure 9 demonstrates the performance of the suggested methodology by varying cluster size and data size. Figure 9A shows various numbers of clustered like 3, 4, 5, 6 representing precision, recall and F-measure are tested. Figure 9 shows performance analysis by varying cluster size and performance analysis by varying data size. Here, the *x* axis represents the cluster size, and *y* axis refers to the corresponding output. When the cluster size is 3, we achieve the precision of 84.89%, recall of 89.90%, and F-measure of 85.63%. The data are partitioned into a number of clusters for easy execution. The performance of the proposed methodology by

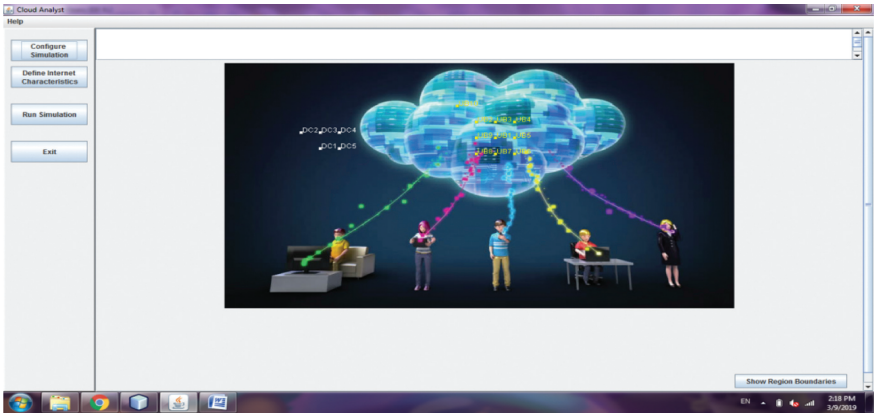

Figure 7: Simulation Started Window.

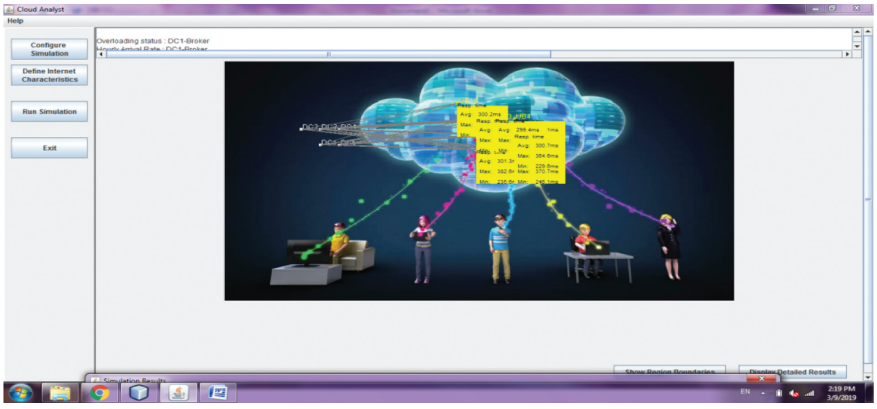

Figure 8: Simulated Window.

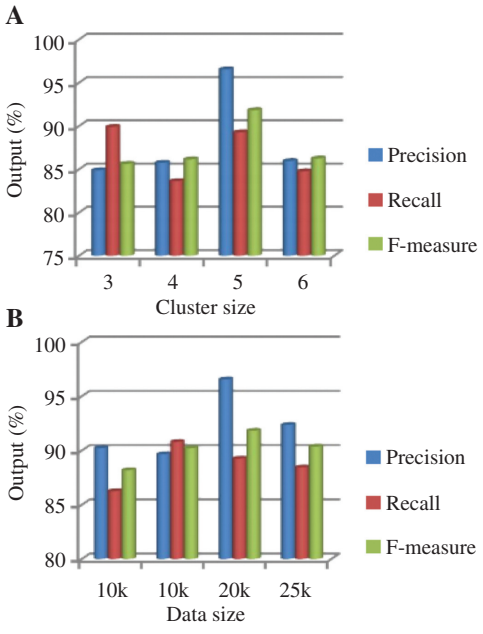

Figure 9: Data size and cluster size performance.
Performance Analysis (A) by Varying Cluster Size and (B) by Varying Data Size.

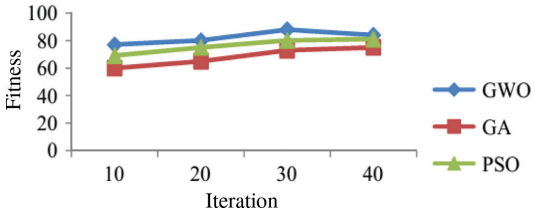

Figure 10: Fitness Comparison.

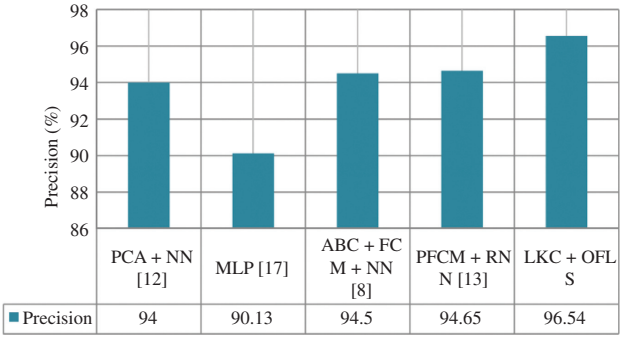

Figure 11: Comparative Analysis Based on the Precision Measure for KDD CUP99 Dataset.

varying data size is afforded in Figure 9B. When the data size is 10,000, the proposed method achieves the maximum precision of 90.22%, recall of 86.26%, and F-measure of 88.1983%. This is because of the proposed LKM and optimal rule generation process. The fitness comparison is afforded in Figure 10. For optimal rule generation, in this paper, GWO is utilized. In Figure 10, two optimization algorithms, namely, PSO and genetic algorithm performances are compared with the proposed GWO. From the result, we can clearly understand that our suggested approach attains the maximum accuracy compared to other works.

4.5 Comparative Analysis Based on Different Clustering Methods

In this section, the performance of the proposed algorithm is analyzed. To prove the effectiveness of the proposed methodology, the proposed LKM algorithm is compared with existing clustering algorithms, namely, k-means clustering and fuzzy means clustering (FCM). The performance analysis is established on cluster sizes and various data sizes.

Table 6 shows the comparative analysis results based on clustering algorithm. When analyzing Table 6, the proposed method attains the average precision of 88.285%, which is 82.72% for using k-means and 83.44% for using FCM based clustering. Moreover, the proposed method attains average recall of 86.89%, which is 74.36% for using k-means and 72.64% for using FCM based clustering. Similarly, compared to F-measure also, we obtain better results. Table 6 shows that our proposed LKM based clustering algorithm is 6.7% better than K-means and 5.8% better than FCM. This because of prototype based hybrid approach. This speeds up the proposed clustering algorithm and overcomes the difficulties present in the K-means clustering algorithm.

Table 6: Performance Analysis by Varying Cluster Size.

Cluster size	Precision			Recall			F-measure		
	LKM+ OFLS	K-means+ OFLS	FCM+ OFLS	LKM+ OFLS	K-means+ OFLS	FCM+ OFLS	LKM+ OFLS	K-means+ OFLS	FCM+ OFLS
3	84.89	80.74	81.46	89.90	74.63	70.875	85.66	80.54	82.63
4	85.75	81.38	82.54	83.63	75.42	72.23	86.88	81.03	83.71
5	96.54	86.93	87.33	89.26	76.00	75.90	91.83	81.10	81.21
6	85.95	81.86	82.44	84.79	71.42	71.58	86.87	81.53	78.94

Table 7 demonstrates the performance of the proposed and existing methods by varying data sizes. Here, the suggested LKM algorithm-based IDS is compared with k-means based IDS and FCM based IDS. Here, the precision value is high when the data size is 20,000; similarly, the precision value is low when the data size is 10,000. Similarly, in this approach, we attain the average recall of 88.69%, which is 75.08% for using k-means based clustering and 76.1% for using FCM based IDS. From the results, it clearly shows that our proposed method is better than the previous clustering algorithm.

4.6 Comparative Analysis Based on the Classifier

In this section the performance of various classifiers based intrusion detection is analyzed. Here, the suggested optimal FLS (OFLS) is compared with a k-nearest neighbor (KNN) based intrusion detection and ANN based intrusion detection. The performance analysis is established on various cluster sizes and different data sizes.

Table 8 shows the comparative analysis based on different classifiers by varying cluster sizes. The precision value is computed by varying the cluster size by 3, 4, 5, and 6. The precision value of cluster size 5 is highly equated with other cluster sizes, and the value is 94.54%. The average recall value of the proposed LKM+OFLS is 86.89% in which the existing LKM+KNN attains 73.44% and the existing LKM+ANN obtains 74.35%. The average F-measure value of the suggested LKM+OFLS is 80.66% in which the existing LKM+KNN obtains 78.46% and the existing LKM+ANN obtains 79.30%. These existing values are low when compared to the suggested LKM+OFLS technique. Due to optimal fuzzy rule selection in FLS, the proposed method attains better result compared to the other method.

4.7 Comparison with Published Papers

To prove the effectiveness of the proposed methodology, in this paper, we compare the performance of our proposed methodology with existing works, namely, PCA+NN [12], MLP [17], ABC+FCM+NN [8], and PFCM+RNN [13]. In [12], a combination of principal component analysis and NN based intrusion detection is made. For feature selection they utilized PCA and for classification they utilized the ANN. In [17], MLP is used for IDS, which is based on off-line analysis approach. The hybridization of a MLP network, ABC, and fuzzy clustering algorithms based IDS is developed in [8]. Similarly, in [13], the IDS is developed based on

Table 7: Performance Analysis by Varying Data Size.

Data size (k)	Precision			Recall			F-measure		
	LKM+ OFLS	K-means+ OFLS	FCM+ OFLS	LKM+ OFLS	K-means+ OFLS	FCM+ OFLS	LKM+ OFLS	K-means+ OFLS	FCM+ OFLS
10	90.22	85.05	85.96	86.26	74.13	75.90	88.16	79.22	80.62
15	89.65	85.64	86.11	90.80	75.54	76.91	90.22	80.28	81.25
20	96.55	86.93	87.33	89.26	76.00	75.90	91.83	81.10	81.21
25	92.36	85.26	86.26	88.45	74.66	75.69	90.36	79.61	80.63

Table 8: Comparative Analysis Based on Different Classifier by Varying Cluster Size.

Cluster size	Precision			Recall			F-measure		
	LKM+ OFLS	LKM+ KNN	LKM+ ANN	LKM+ OFLS	LKM+ KNN	LKM+ ANN	LKM+ OFLS	LKM+ KNN	LKM+ ANN
3	84.89	83.95	84.54	89.92	73.22	73.78	85.66	78.22	78.79
4	85.71	84.33	85.02	83.63	73.38	74.37	86.88	78.43	79.34
5	96.54	85.05	85.96	89.26	74.13	75.90	91.83	79.22	80.62
6	85.99	83.66	84.26	84.79	73.03	73.36	86.87	80.62	78.43

possibilistic PFCM with RNN. To compare these methods, NSL-KDD cup 99 dataset is utiized. Comparative analysis based on the precision measure for KDD CUP99 dataset is given in Figure 11.

OFLS+LKM based IDS is explained in this paper. Here, for clustering process, LKM is utilized, and for intrusion detection, optimal NN is utilized. When analyzing Figure 11, we obtain the average maximum accuracy of 96.54%, which is 94% for using PCA-NN [12], 90.13% for using MLP based IDS [17], 94.5% for using [8], and 94.65 for using PFCM+RNN [13]. This is because of LKM and weight optimization process. From the result, we clearly understand that our proposed approach is better compared to other approaches.

5 Conclusion

Nowadays, in the cloud, system security is one of the major worries because of various attacks and vulnerabilities. As a result, attack detection is an imperative segment in system security. In this paper, a combination of FLS, GWO, and LKM generates a novel IDS which is presented. At various trainings, subsets are developed by LKM method. The discrimination among normal and abnormal data is done by the FLS. The optimal rules are generated applying GWO algorithm. The experimental results applying the KDD CUP 1999 dataset shows the effectiveness of our approach, which provides better precision than the existing method. In the future, we will develop the security of the data applying cryptographic algorithms.

Bibliography

[1] S. Alam, M. Shuaib and A. Samad, A collaborative study of intrusion detection and prevention techniques in cloud computing, in: *International Conference on Innovative Computing and Communications*, pp. 231–240, Springer, Singapore, 2019.

[2] M. Baykara and R. Das, A novel honeypot based security approach for real-time intrusion detection and prevention systems, *J. Inform. Secur. Appl.* **41** (2018), 103–116.

[3] K. Bhushan and B. B. Gupta, Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment, *J. Amb. Intel. Hum. Comput.* **10** (2018) 1–13.

[4] S. Chavan, K. Shah, N. Dave, S. Mukherjee, A. Abraham and S. Sanyal, Adaptive neuro-fuzzy intrusion detection systems, in: *International Conference on Information Technology: Coding and Computing*, Proceedings, ITCC 2004, vol. 1, pp. 70–74, IEEE, Las Vegas, NV, USA, 2004.

[5] F. Feng, X. Liu, B. Yong, R. Zhou and Q. Zhou, Anomaly detection in ad-hoc networks based on deep learning model: a plug and play device, *J. Ad Hoc Netw.* **84** (2019), 82–89.

[6] P. Ghosh, A. Karmakar, J. Sharma and S. Phadikar, CS-PSO based intrusion detection system in cloud environment, in: *Emerging Technologies in Data Mining and Information Security*, pp. 261–269, Springer, Singapore, 2019.

[7] A. Girma, M. Garuba and R. Goel, Advanced machine language approach to detect DDoS attack using DBSCAN clustering technology with entropy, in: *Information Technology – New Generations*, pp. 125–131, Springer, Cham, 2018.

[8] B. Hajimirzaei and N. J. Navimipour, Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm, *J. ICT Exp.* **5** (2018), 56–59.

[9] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson and X. Bellekens, A taxonomy and survey of IDS design techniques, network threats and datasets, *Assoc. Comput. Mach.* **1** (2018), 1.

[10] P. Kachurka and V. Golovko, Neural network approach to real time network intrusion detection and recognition, in: *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Prague, pp. 15–17, September 2011.

[11] R. Kozik, M. Choraś, M. Ficco and F. Palmieri, A scalable distributed machine learning approach for attack detection in edge computing environments, *J. Parallel Distr. Comput.* **119** (2018), 18–26.

[12] Z. Mahmood, C. Agrawal, S. S. Hasan and S. Zenab, Intrusion detection in cloud computing environment using neural network, Int. J. Res. Comput. Eng. Electron. **1** (2014).

[13] M. Manickam, N. Ramaraj and C. Chellappan, A combined PFCM and recurrent neural network based IDS for cloud environment, *Int. J. Bus. Intel. Data Min.* **1** (2017), 504–527.

[14] M. Mazini, B. Shirazi and I. Mahdavi, Anomaly network-based IDS using a reliable hybrid artificial bee colony and AdaBoost algorithms, *J. King Saud Univ. Comput. Inform. Sci.* **31** (2019), 541–553.

[15] P. Mell and T. Grance, *Effectively and securely using the cloud computing paradigm*, vol. 2, pp. 304–311, NIST, Information Technology Laboratory, Gaithersburg, MD, USA, 2009.

[16] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *J. Adv. Eng. Softw.* **69** (2014), 46–61.

[17] M. Moradi and M. Zulkernine, A neural network based system for intrusion detection and classification of attacks, in: *Proceedings of the IEEE International Conference on Advances in Intelligent Systems-Theory and Applications*, pp. 15–18, Luxembourg-Kirchberg, Luxembourg, 2004.
[18] S. Ramteke, R. Dongare and K. Ramteke, IDS for cloud network using FC-ANN algorithm, *Int. J. Adv. Res. Comput. Commun. Eng.* **2** (2013).
[19] T. H. Sarma, P. Viswanath and B. E. Reddy, A hybrid approach to speed-up the k-means clustering method, *Int. J. Mach. Learn. Cyb.* **4** (2013), 107–117.
[20] A. S. Sohal, R. Sandhu, S. K. Sood and V. Chang, A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments. *Comput. Secur.* **74** (2018), 340–354.
[21] F. Song, Z. Guo and D. Mei, Feature selection using principal component analysis, in: *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM)*, vol. 1, no., pp. 27–30, Yichang, China, 12–14 Nov. 2010.
[22] A. N. Toosi and M. Kahani, A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers, *J. Comput. Commun.* **30** (2007), 2201–2212.
[23] A. T. Velte, T. J. Velte and R. C. Elsenpeter, *Cloud computing: a practical approach*, p. 44, McGraw-Hill, New York, 2010.
[24] G. Wang, J. Hao, J. Ma and L. Huang, A new approach to intrusion detection using artificial neural networks and fuzzy clustering, *J. Expert Syst. Appl.* **37** (2010), 6225–6232.