Using Artificial Intelligence Techniques For Intrusion Detection System

Manar Y. Ahmed **Bayda I. Khaleel**

College of Computer Sciences and Mathematics
University of Mosul, Mosul, Iraq

Received on: 16/10/2012 **Accepted on: 30/01/2013**

ABSTRACT

Along with the development and growth of the internet network, and the rapid expansion of World Wide Web and local network systems have changed the computing world in the last decade. Nowadays, as more people make use of the internet, their computers and the valuable data in their computer system contain become more exposed to attackers. Therefore, there is an increasing need to protect computer and network from attacks and unauthorized access. Such that network intrusion classification and detection systems to prevent unlawful accesses. This work has taken the advantage of classification and detection abilities of Artificial Intelligent Techniques AITs algorithms to recognize intrusion(attack) and also detect new attacks. These algorithms are used to multi classifier and binary classifier for network intrusion and detect it, AITs such as unsupervised and supervised fuzzy clustering algorithms (Fuzzy C-Mean FCM, Gustafson-Kessel GK, and Possibilistic C-Means PCM), was applied to classify intrusion into 23 classes according to the subtype of attack. The same dataset classifies it into 5 classes according to the type of attacks (Normal, DoS, Probe, U2R, R2L). And also classifies this dataset into 2 classes (Normal, and Attack), one for normal traffic and another for attack, also these algorithms are used to detect intrusion.

Other techniques were used which are artificial neural network (ANN) represented by counter propagation neural network (CPN) which is hybrid learning (supervised and unsupervised) that is applied to classify intrusion into 23, 5 and 2 class(es) and used it to detect the network intrusions, and then we combined fuzzy c-mean with two layers Kohonen layer and Grossberg layer for counter propagation neural network to produce the proposed approach or system that called it fuzzy counter propagation neural network (FCPN) were applied it to classify network intrusion into 23, 5 and 2 class(es) and detect the intrusion. DARPA 1999 (Defense Advanced Research Project Agency) dataset which is represented by Knowledge Discovery and Data mining (KDD) cup 99 dataset was used for both training and testing. This research evaluates the performance of the approaches that are used that obtained high classification and detection rate with low false alarm rate. The performance of the proposed approach FCPN is the best if it is compared with the other approaches that are used and with previous works. Finally, in this research comparisons are made between the results obtained from the application of these algorithms on this dataset and the FCPN is the best approach that is implemented into Laptop where, CPU 2.27GH and RAM are 2.00 GB.

Keyword: Intrusion Detection, Unsupervised and Supervised (Fuzzy C-Means(FCM), Possibilistic C-Means(PCM) and Gustafson-Kessel (GK)) algorithms, Fuzzy Counter Propagation Neural Network (FCPN), Kdd Cup 99 Data Set.

استخدام التقنيات الذكائية الاصطناعية لنظام كشف التطفل

بيداء ابراهيم خليل **منار يونس كشمولة**

كلية علوم الحاسبات والرياضيات، جامعة الموصل

تاريخ استلام البحث: 2012/10/16 **تاريخ قبول البحث: 2013/01/30**

الملخص

مع التطور والنمو الكبير لشبكة الانترنيت، والتوسع السريع للشبكة العنكبوتية العالمية وأنظمة الشبكة المحلية، تغير عالم الحاسوب في الآونة الأخيرة. ففي يومنا هذا الكثير من الناس يستخدمون الانترنيت والحواسيب، وقيم البيانات التي تحويها أنظمة هذه الحاسبات والتي أصبحت أكثر استغلالا من قبل المهاجمين. لذلك زادت الحاجة لأنظمة الحماية مثل أنظمة كشف وتصنيف التطفل لحماية الحاسبة والشبكة من الهجمات والوصول الغير مخول به. وهنا تم أخذ الفائدة من قابليات التصنيف والكشف لخوارزميات التقنيات الذكائية الاصطناعية لتصنيف التطفل وكشف الهجوم الجديد. وخوارزميات التقنيات الذكائية هذه أُستخدمت للتصنيف المتعدد والثنائي لتطفل الشبكة وكشفه. مثل خوارزميات العنقدة المضببة الإرشادية وغير الإرشادية (FCM, GK, PCM, SFCM, SGK, SPCM) والتي أُستخدمت لتصنيف التطفل إلى 23 صنف طبقا لاسم الهجمة التابع لنوع الهجوم الرئيسي، وكذلك طُبقت هذه الخوارزميات لتصنيف التطفل إلى 5 أصناف طبقا لنوع الهجوم الرئيسي، وصنفت نفس البيانات إلى صنفين أحدهما للمرور الطبيعي والآخر للهجوم، وأُستخدمت هذه الخوارزميات أيضاً لكشف التطفل.

أُستخدمت تقنيات ذكائية اصطناعية أُخرى متمثلة بشبكة الـ CPN ذات التعليم المهجن الإرشادي وغير الإرشادي والتي طبقت لتصنيف التطفل إلى 23، 5، 2 صنف وكشفه، ومن ثم تم دمج خوارزمية الـ FCM مع الطبقتين لشبكة الـ CPN، طبقة كوهين وطبقة كروس بيرج لينتج نظام جديد أو طريقة مقترحة سميت FCPN وطُبقت هذه الطريقة لتصنيف التطفل إلى 23، 5، 2 صنف وكشف التطفل. وأُخذت بيانات التدريب والاختبار من DARPA والمتمثلة ببيانات الـ KDD CUP 99. وتم تقييم أداء الطرائق المستخدمة والتي حصلت على أعلى نسبة تصنيف وكشف واقل نسبة إنذار كاذب. وأداء الطريقة الجديدة هو الأفضل مقارنة مع الطرق الأخرى التي استخدمت في هذا العمل وكذلك مقارنة مع الأعمال السابقة. وأخيرا تمت مقارنة النتائج التي تم الحصول عليها بعد تطبيق الخوارزميات على هذه البيانات والتي نفذت على حاسبة من نوع أج بي سرعة وحدة المعالجة المركزية هي 2.27 كيكا هيرتز والذاكرة 2.00 كيكا بايت.

الكلمات المفتاحية: كشف التطفل، خوارزميات العنقدة المضببة الارشادية وغير الارشادية (FCM, GK, PCM, SFCM, SGK, SPCM) ، شبكة الـ CPN ذات التعليم المهجن، بيانات الـ KDD CUP 99 .

1. General Introduction

Network security is fast becoming an absolute necessity to protect information contained in the computer systems world wide. And with the rapid expansion of computer networks during the past decade[1], and the network grows in size and complexity and computer services expansions, vulnerabilities within local area and wide area network has become mammoth albeit problematic. The problems occur due to the increasing number of intrusion tools and exploiting scripts which can entice anyone to launch an attack on any vulnerable machines. The attack can be launched in term of fast attack or slow attack. Fast attack can be defined as an attack that uses a large amount of packet or connection within a few seconds. Meanwhile, slow attack can be defined as an attack that takes a few minutes or a few hours to complete. Both of the attacks give a great impact to the network environment due to the security breach[2]. The number of intrusion in computer networks has grown extensively, and many new hacking tools and intrusive methods have appeared which attackers are used[3]. Intrusion detection techniques can be categorized into misuse detection and anomaly detection .

- Misuse detection uses the patterns of well-known attacks or vulnerable spots in the system to identify intrusions [4]. Misuse detection is based on the knowledge of system vulnerabilities and known attack patterns. Misuse detection is concerned with finding intruders who are attempting to break into a system by exploiting some known vulnerability, ideally, a system security administrator should be

aware of all the known vulnerabilities and eliminate them [5].

- Anomaly detection attempts to determine whether can be flagged as intrusions.

There are three types of intrusion detection systems: Host-based Intrusion Detection System (HIDS), Network-based Intrusion Detection System (NIDS), and combination of both types (Hybrid Intrusion Detection System) [6] and [4].

2. KDD Cup 99 Dataset

Since 1999, (Knowledge Discovery and Data Mining) KDD'99 has been the most wildly used dataset. The network data is distributed by MIT Lincoln Lab for DARPA[3][4]. This dataset is built based on the data captured in the Department of DARPA'98 IDS evaluation program. DARPA'98 is about 4 gigabytes of compressed raw tcpdump data of 7 weeks of training set and two weeks of test data. It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data which makes the task more realistic. The "10% KDD" datasets contain a total number of 23 training attack ,with additional 15 types in the test data only which contains 38 attacks in "Corrected KDD", recorded connection in KDD data are a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. The KDD cup 99 dataset includes a set of 41 features derived for each connection and a label which specifies the status of connection records as either normal or specific attack type[7][8].Attack type falls into four main categories [4][9] and [10]:

- Denial of Service(DOS) attacks, which prevent a computer from complying with legitimate requests by consuming its resources.
- Probe attack, which are scanning and polling activities that gather information on vulnerabilities for future attack.
- Remote-to-Local(R2L) attack, which are local non-authorized access attempts from a remote machine.
- User-to-Root(U2R) attack, which have the goal of obtaining illegal or non-authorized super-user or root privileges.

The total number of connection records in training dataset is kdd 10% dataset (494020) records . And the total number of connection records in testing dataset is kdd corrected dataset (311029) records. This dataset consists of symbolic and numeric values, all symbolic values were transformed into numeric values [11] such as three types of protocols (tcp, udp, icmp) and 68 types of services and 11 types of flag, each one takes value from [1..N] and then normalized all input data of 10%kdd dataset[12].

Table (1). Basic Characteristics of the KDD 99 Intrusion Detection Dataset in Terms of Number of Samples[13]

Dataset	Normal	DoS	Probe	U2R	R2L	Total
"Corrected KDD"	60593	229853	4166	70	16347	311029
"10% KDD"	97277	391458	4107	52	1126	494020

3. Preprocessing Dataset

From the KDD Cup 99 intrusion detection dataset, 41 features were derived to summarize each connection information. In order to train an architecture, several data of enumeration and normalization operations were necessary. As a first approach, symbolic variables in the dataset were enumerated and all variables were normalized. Thus, each instance of a symbolic feature was first mapped to sequential integer values.

This dataset consists of symbolic and numeric values, all symbolic values were transformed into numeric values such as three types of protocols (tcp, udp, icmp) and 68 types of services in KDD cup 99 and 11 types of flag, each one takes value from [1..N] as described in table (2) and in figures (1) and (2) [14], and each numerical value in the dataset is normalized between 0.0 and 1.0 according to the following equation :

$$X = \frac{x - \min}{\max - \min} \quad \ldots(1)$$

Where, X is the numerical value, $\min$ is the minimum value for the attribute that x belongs to, and $\max$ is the maximum value for the attribute that x belongs to[15].

Table (2). Numerical Values of KDD Dataset Features

Protocol Type	Feature Value	Services	Feature Value	Services	Feature Value	Services	Feature Value	Flage	Feature Value
tcp	1	Private	1	whois	24	shell	48	SF	1
udp	2	Smtp	2	time	25	efs	49	SH	2
icmp	3	http	3	mtp	26	login	50	SO	3
		ftp_data	4	gopher	27	printer	51	S1	4
		X11	5	remot_job	28	netbios_ssn	52	S2	5
		IRC	6	rje	29	csnet_ns	53	S3	6
		Telent	7	link	30	nntp	54	RSTR	7
		Domain	8	ctf	31	supdup	55	REJ	8
		Finger	9	hostname	32	http_443	56	RSTO	9
		Other	10	iso_tsap	33	uucp_path	57	OTH	10
		ftp	11	pop_2	34	domain_u	58	RSTOSO	11
		Auth	12	netbios_dgm	35	ntp_u	59		
		Imap4	13	netbios_ns	36	ecr_i	60		
		pop_3	14	sql_net	37	eco_i	61		
		Sunrpc	15	bgp	38	tim_i	62		
		pm_dump	16	vmnet	39	urh_i	63		
		Echo	17	Z39_50	40	icmp	64		
		Discard	18	ldap	41	urp_i	65		
		Systat	19	nnsp	42	red_i	66		
		Daytime	20	kshell	43	tftp_u	67		
		Netstat	21	klogin	44	harvest	68		
		Ssh	22	uucp	45				
		Name	23	courier	46				
				exec	47				

```
0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00
,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.
0,tcp,smtp,SF,751,279,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,
28,19,0.68,0.14,0.04,0.00,0.00,0.00,0.00,0.00,normal.
0,tcp,finger,SF,9,140,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,3
8,5,0.13,0.11,0.03,0.00,0.00,0.00,0.00,0.00,normal.
0,udp,domain_u,SF,33,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,0.00,0.00,0.00,0.00,1.00,0.00,1.
00,78,14,0.18,0.06,0.18,0.00,0.00,0.00,0.00,0.00,normal.
0,icmp,eco_i,SF,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1
,1,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,normal.
```

Figure 1. The Original Data of KDD Cup 99

```
0,1,3,1,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0,0,0,0,1,0,0,9,9,1,0,0.11,0,0,0,0,0,1
0,1,2,1,751,279,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,28,19,0.68,0.14,0.04,0,0,0,0,0,1
0,1,9,1,9,140,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,38,5,0.13,0.11,0.03,0,0,0,0,0,1
0,2,58,1,33,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,0,0,0,0,1,0,1,78,14,0.18,0.06,0.18,0,0,0,0,0,1
0,3,61,1,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,1,1,1,0,1,0,0,0,0,0,1
```

Figure 2. After Transform from Original Data

4. Performance Measures

The indicators were used to measure the accuracy of the IDS[16]:

True positive(TP): classifying an intrusion as intrusion. The true positive rate is synonymous with detection rate, sensitivity and recall which are other terms often used in the literature.

False positive(FP): incorrectly classifying normal data as an intrusion . Also is known as a false alarm.

True negative(TN): correctly classifying normal data as normal. The true negative rate is also referring to specificity.

False negative(FN): incorrectly classifying an intrusion as normal[17].

The performance metrics calculated from these are:

$$True\ Positive\ rate\ (TPR) = \frac{TP}{TP + FN} = \frac{\#\ correct\ \text{int}\ rusions}{\#\ \text{int}\ rusions} \times 100 \quad \ldots(2)$$

$$False\ Positive\ rate\ (FPR) = \frac{FP}{TN + FP} = \frac{\#\ normal\ as\ \text{int}\ rusions}{\#\ normal} \times 100 \quad \ldots(3)$$

$$True\ negative\ rate\ (TNR) = \frac{TN}{TN + FP} = \frac{\#\ correct\ normal}{\#\ normal} \times 100 \quad \ldots(4)$$

$$False\ negative\ rate\ (FNR) = \frac{FN}{TP + FN} = \frac{\#\ \text{int}\ rusions\ as\ normal}{\#\ \text{int}\ rusions} \times 100 \quad \ldots(5)$$

And over all classification rate is also referred to as accuracy can be calculated as follows[18] and [17]

$$classification\ rate = \frac{number\ of\ samples\ classified\ correctly}{number\ of\ samples\ used\ for\ training} \times 100 \quad \ldots(6)$$

$$Detection\_rate = \frac{number\ of\ correctly\ \text{det}\ ected\ samples}{total\ number\ of\ samples} \times 100 \quad \ldots(7)$$

5. Clustering

We are living in a world full of data. Every day, people encounter a large amount of information and store or represent it as data, for further analysis and management. One of the vital means in dealing with these large data is to classify or group them into a set of categories or clusters. Clustering is the process of grouping a dataset in such a way that the similarity between data within a cluster is maximized, while the similarity between data of different clusters is minimized. Clustering or classification systems are either supervised or unsupervised, unsupervised clustering takes an unlabelled set of data and partition it into groups of examples, without additional knowledge. Supervised clustering , on the other hand, assumes that the class structure is already known. It takes a set of examples with class labels[19].

6. Unsupervised Fuzzy Clustering Algorithms

6.1 Fuzzy C-Means (Fcm)Algoritm

The most popular fuzzy clustering algorithm is fuzzy c-means (Bezdek). It is a data clustering technique, wherein each data point belongs to a cluster to some degree that is specified by a membership grad[20]. It is based on minimization of the objective function as in equation (8) [21]:

$$J_m(\mu, v) = \sum_{k=1}^{N} \sum_{i=1}^{c} (\mu_{ki})^m\ d_{ik}(x_k, v_i) \quad \ldots(8)$$

Where c and m are user-defined parameters and represent the number of clusters and fuzzification factors, respectively, N denotes the number of patterns, conventional FCM algorithm includes the following steps:

1. Initialize the cluster center $V=\{v_1,\ldots v_i,\ldots v_c\}$, or initialize the membership matrix μ_{ki} and, then calculate the centers.
2. calculate the fuzzy membership μ_{ki}, using

$$\mu_{ki}=\left[\sum_{j=1}^{c}\left(\frac{d_{ki}}{d_{kj}}\right)^{\frac{2}{m-1}}\right]^{-1} \quad \ldots(9)$$

   where, $d_{ki}=\| x_k - v_i \|$, i= 1, …., n, j=1, … , c.
3. compute the fuzzy centers v_i by using

$$v_i=\frac{\sum_{k=1}^{N}(\mu_{ki})^m X_k}{\sum_{k=1}^{N}(\mu_{ki})^m} \quad \ldots(10)$$

4. Repeat steps (2) and (3) until the minimum J value is achieved.
5. Finally, defuzzification is necessary to assign each data point to a specific cluster(i.e. by setting a data point to a cluster for which the degree of the membership is maximal).

6.2 Gustafson-Kessel(Gk) Algorithm

The Gustafson-kessel is an extension of the fuzzy c-means algorithm[22]. It used mahalanobis distance. The objective function is:

$$J_q(S,\mu,V)=\sum_{j=1}^{n}\sum_{i=1}^{c}(\mu_{ij})^q D_{ij}^2 \quad \ldots(11)$$

The various steps involved in the GK algorithm are given below[23]:

1. Fix fuzzifier , and threshold ε.
2. Initialize membership values μ_{ij}.
3. For i=1, 2, …, max iteration
4. Update the values of clusters v_i by using equation (12)

$$v_i=\frac{\sum_{j=1}^{n}(\mu_{ij})^q S_j}{\sum_{j=1}^{n}(\mu_{ij})^q} \quad \ldots(12)$$

5. Calculate the covariance matrices by using equations (13) and (14)

$$A_i=\left[\rho_i \det(F_i)^{\frac{1}{n}} F^{-1}\right] \quad \ldots(13)$$

$$F_i=\frac{\sum_{j=1}^{n}(\mu_{ij})^q (S_j - v_i)^T (S_j - v_i)}{\sum_{j=1}^{n}(\mu_{ij})^q} \quad \ldots(14)$$

6. Calculate the distance norms by using equation (15):
$$D_{ij}^2 = (S_j - v_i)^T A_i (S_j - v_i) \quad \ldots(15)$$
7. Update μ_{ij} by using equation (16)
$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{D_{ij}}{D_{kj}} \right)^{\frac{2}{q-1}}} \quad \ldots(16)$$
8. IF $\| \mu_{ij}^I - \mu_{ij}^{I-1} \| \leq \varepsilon$, then stop End for

6.3 Possibilistic C-Meams (Pcm) Algorithm

The possibilistic c-means (PCM) algorithm is based on a modification of the objective function of (FCM). The objective function is:
$$\min \left\{ J_m(x, \mu, c) = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_{ij}^m d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{N} (1 - \mu_{ij})^m \right\} \quad \ldots(17)$$
where, dij is given by $\| x_j - v_i \|$.
the steps of (PCM) algorithm are seen below[24]:

1. Initialize the cluster center V= {v_1,…v_i,…v_c}, or initialize the membership matrix μ_{ki} and, then calculate the centers.
2. calculate the fuzzy membership μ_{ki} by using
$$\mu_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i} \right)^{\frac{1}{m-1}}} \quad \ldots(18)$$
Where η_i is the suitable positive number .
3. compute the fuzzy centers v_i by using
$$v_i = \frac{\sum_{j=1}^{N} (\mu_{ij})^m X_j}{\sum_{j=1}^{N} (\mu_{ij})^m} \quad \ldots(19)$$
4. Repeat steps (2) and (3) until the minimum J value is achieved.

7. Supervised Fuzzy Clustering Algorithms

7.1 Supervised Fuzzy C-Means(Sfcm)Algoritm

Class labels always provide a useful guidance during training process, as being done in all the learning methods. Hence, it becomes necessary to use the labeled samples in training phase and unlabeled samples in testing phase to improve the performance of FCM. This idea led to the development of a new algorithm called 'Supervised Fuzzy C-Means' algorithm, a slight modification of FCM(Hong-Bin). The SFCM clustering technique aims to develop classifiers that can utilize both labeled and unlabeled samples. The objective function of the SFCM is defined as:
$$J_m(U, v) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^m d_{ik}^2 + a \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik} - f_{ik})^m d_{ik}^2 \quad \ldots(20)$$

μ_{ik} Membership degree of k^{th} data point belonging to the i^{th} cluster.

f_{ik} Membership degree of k^{th} labeled sample belonging to the i^{th} cluster.

The coefficient 'a' denotes scaling factor and 'm' denotes the fuzzy coefficient. The role of 'a' is to maintain a balance between supervised and unsupervised component within the optimization mechanism and parameter 'm' controls the amount of fuzziness in the classification. The a=L/n, L denoting the size of labeled samples[25]. The steps in this algorithm are as follows:

1. Fix the number of clusters c. Initialize membership values of matrix F of size c × n with 0 or 1 in accordance with class labels. Initialize fuzzy partition matrix $U^{(0)}$ with random values between 0 and 1.
2. Start the iterative procedure and set the iteration count, t=1.
3. Calculate the clusters (prototype) of the clusters by using equation (21) given below

$$v_{ij}^{(t)} = \frac{\sum_{k=1}^{n} (U_{ik}^{(t-1)})^m \, Z_{kj}}{\sum_{k=1}^{n} (U_{ik}^{(t-1)})^m} \qquad \ldots(21)$$

4. Calculate the distance, $d_{ik}^{(t)}$, between i^{th} cluster center and k^{th} dataset. The distance measure used is Euclidean Distance as given by equation(22).

$$d_{ik}^{(t)} = \sqrt{\sum_{j=1}^{m} (Z_{kj} - v_{ij}^{(t)})^2} \qquad \ldots(22)$$

5. Update the fuzzy partition matrix, $U^{(t+1)}$, for the next iteration as follows:

$$\mu_{ik}^{(t+1)} = (1-a)\left[\sum_{j=1}^{c}\left(\frac{d_{ik}^{(t)}}{d_{jk}^{(t)}}\right)^{\frac{2}{m-1}}\right]^{-1} + a\, f_{ik} \qquad \ldots(23)$$

6. if $\| U^{(t+1)} - U^{(t)} \| \leq \varepsilon$ (ε being iterative accuracy), stop the iteration and output v (cluster center), U (fuzzy matrix); else increment the iteration count, and return to step 3.

7.2 Supervised Gustafson-Kessel (Sgk) Algorithm

At the same of algorithm of the FCM that is modified by (Hong-Bin) to SFCM was explained above in section(7.1). We have modified the unsupervised Gustafson-Kessel (GK) algorithm to supervised Gustafson-Kessel (SGK) by adding two parameters 'a' and 'f' to equation fuzzy membership μ_{ij} in equation number (16) to be as shown in the equation (24) with the same steps of algorithm were used.

$$\mu_{ij} = (1-a)\left[\frac{1}{\sum_{k=1}^{c}\left(\frac{D_{ij}}{D_{kj}}\right)^{\frac{2}{q-1}}}\right] + a\, f_{ij} \qquad \ldots(24)$$

Where, $D_{ij}^2 = (S_j - v_i)^T \, A_i \, (S_j - v_i)$

7.3 Supervised Possibilistic C-Meams (Pcm) Algorithm

The same as algorithm of the FCM that is modified by (Hong-Bin) to SFCM as explained above in section(7.1), We have modified the unsupervised possibilistic c-means (PCM) algorithm to a supervised possibilistic c-means (SPCM) by adding two parameters 'a' and 'f' to equation fuzzy membership μ_{ij} in the equation (18) to be as shown in (25).

$$\mu_{ij} = (1-a)\left[\frac{1}{1+\left(\frac{d_{ij}^2}{\eta_i}\right)^{\frac{1}{m-1}}}\right] + a\, f_{ij} \qquad \ldots(25)$$

Where, dij is given by $\| x_j - \mathrm{v}_\mathrm{i} \|$.

8. Counterpropagation Network

The CP network was first developed by Hecht-Nielsen [26], and consisted of combining the Kohonen network with a Grossberg layer [27]. The general form of the CP network can be seen in figure (1). The input nodes of the Kohonen layer are connected to the Kohonen neurons by weights w_{ij}, while the Kohonen outputs are connected to the Grossberg layer by the connecting weights v_{ij} [28]. The learning of CPN can be split into two stages, unsupervised and supervised. Unsupervised learning is used during the first stage for clustering the input vectors to separate distinct sets of input data. During the second stage of learning, the weight vector between the Kohonen and Grossberg layers are adjusted by supervised learning to reduce the errors between the CPN outputs and the corresponding desired targets. During the

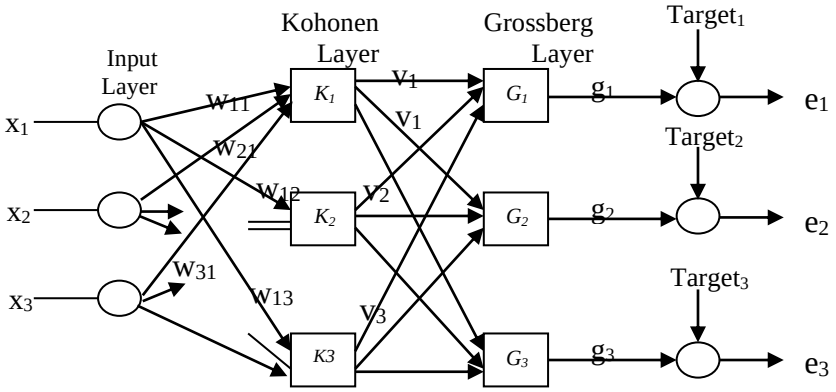

Figure 3. counter propagation network[27][28]

First stage, the distances between the input vector $x=(x_1, ..., x_i, x_n)^T$ composed of input nodes and all of the j Kohonen nodes with n dimensions are determined to compete for the winner.

The training steps of the counter propagation network (CPN) [29] and [30] as follows:

1. A vector pair (x, y) of the training set, is selected in random.
2. Normalize the input vector x to obtain x' by the equation (26):

$$x' = \frac{x_i}{\sqrt{\sum_j x_j^2}} \qquad \ldots(26)$$

3. the weights are obtained as equation (27)
$$w = x' \quad \ldots(27)$$
namely, the weight vector of the wining Kohonen neuron(the j th neuron in the Kohonen layer) equals(best approximates) the input vector.
4. In the hidden competitive layer, the distance between the weight vector and the current input vector is calculated for each hidden neuron j according to the equation(28)
$$D_j = \sqrt{\sum_{i=1}^{k} (x_j - w_{ij})^2} \quad \ldots(28)$$
where, k is the number of the hidden neurons and w_{ij} is the weight of the synapse that joins the i th neuron of the input layer with the j th neuron of the Kohonen layer.
5. The winner neuron W of the Kohonen layer is identified as the neuron with the minimum distance value D_j.
6. The synaptic weights between the winner neuron W and all neuron of the input layer are adjusted according to the equation (29)
$$w(t+1) = w(t) + \alpha[x - w(t)] \quad \ldots(29)$$
where α coefficient is known as the Kohonen learning rate.
7. The weight between Kohonen layer and Grossberg layer v_{ij} obtained at the same way to obtain w_{ij} weight between input layer and Kohonen layer as in equation (27) above.
8. Obviously, only weights from non-zero Kohonen neurons (non-zero Grossberg layer inputs) are adjusted. Weight adjustment as follows:
$$v_{ij}(t+1) = v_{ij}(t) + \beta[T_i - v_{ij}(t)\, k_j] \quad \ldots(30)$$
T_i being the desired outputs(targets), β is small number that represented the learning rate of Grossberg layer.
9. A major asset of the Grossberg layer is the ease of its training. First the output of the Grossberg layer is calculated as in equation (31)
$$g_i = \sum v_{ij} k_j = v_{ih} k_h = v_{ih} \quad \ldots(31)$$
k_j being the Kohonen layer outputs and v_{ij} denoting the Grossberg layer weights.

9. Hybrid Counterpropagation Network With Fcm

Counterpropagation developed by Hecht-Nielsen can be generalized to design a Fuzzy counterpropagation network, by extending the two layers (Kohonen's layer and Grossberg's layer) to a fuzzy counterpropagation network. The basic objective of this network is to cluster the input patterns, in each a way that total Euledian distance between each pattern and its nearest cluster centroid is minimum in Kohonen layer, and we take the minimum distance output for each winner neuron in Kohonen layer and maximum output neuron in Grossberg layer. A novel method is proposed in this research by using fuzzy c-means algorithm in Grossberg layer which is called FCPN, and steps (4 and 5) in the following algorithm were used to implement the above algorithm which has been applied by using kdd 99 dataset. The algorithm for fuzzy counterpropagation is shown below.

1. A vector pair (x, y) of the training set, is selected randomly. It is normalized and used as an input to obtain the weight by the equation (26) and (27) respectively.
2. Compute the distances $d(x_k, w_i)$ from the input pattern x_k to each of the competing neurons w_i.
3. Compute the membership of the winner neuron based on the distance measure $d(x_k, w_i)$.
4. Update the weight associated with each neuron. The weight updation is performed in accordance to the following rule.
   $$w_i(t+1) = w_i(t) + \alpha\, z_i(t)\left[x_k - w_i(t)\right] \quad \ldots(32)$$
   where, z_i is the fuzzy scaling function given by:
   $$z_i = (\mu_{ik})^m$$
   where,
   $$\mu_{ik} = \left[\sum_{p}^{c}\left(\frac{D_{ik}}{D_{pk}}\right)^{\frac{2}{m-1}}\right]^{-1} \quad \ldots(33)$$
   and $D_{ik} = d(x_k, w_i)$. The scaling function z_i depends on the fuzzy generator m which is a real number greater than 1.
5. Compute the membership between the winner neuron and Grossberg layer based on the distance measure $d(k_j, v_i)$. And update the weight associated with each neuron. The weight updation is performed in accordance to the following rule.
   $$v_{ij}(t+1) = v_{ij}(t) + \beta\, z_i(t)\left[T_i - v_{ij}(t)k_j\right] \quad \ldots(34)$$
   where z_i is the fuzzy scaling function given by:
   $$z_i = (\mu_{ij})^m$$
   Where,
   $$\mu_{ij} = \left[\sum_{p}^{c}\left(\frac{D_{ij}}{D_{pj}}\right)^{\frac{2}{m-1}}\right]^{-1} \quad \ldots(35)$$
   and $D_{ij} = d(k_j, v_i)$. The scaling function z_i depends on the fuzzy generator m which is a real number greater than 1.
6. Calculate the output of Grossberg as equation (31).

The CPN and FCPN used for the classification and detection network intrusion. These two methods (CPN, and FCPN) performed binary classifier and multi classifier for the dataset. Figure (4) shows the system designed of these two methodes for binary classifying. The system used the input dataset (normal and attack) that contains 41 features, which are equal to nodes in the input layer. While, in the Kohonen or clustering layer, there are 2 Kohonen nodes, one for normal and the other for attack. Finally the number of the output node in the output layer is 2 according to the target output.

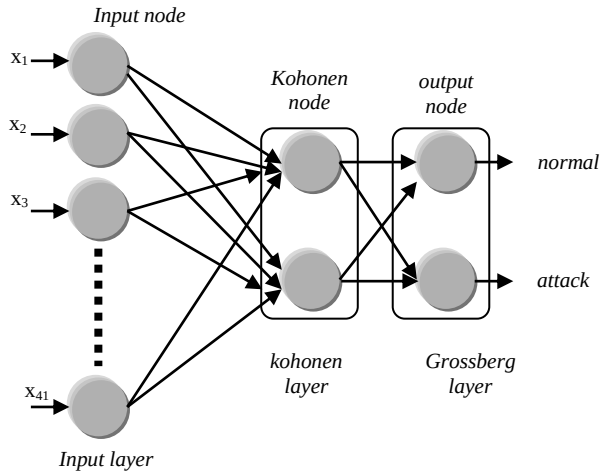

Figure 4. CPN and FCPN Architecture for Binary Classification Network Intrusion

Figure (5) shows the system architecture of CPN and FCPN for multi classifier. The system uses the same input dataset, so, there is 41 nodes in input layer and 5 nodes in Kohonen layer. The last layer consists of 5 output nodes in output layer, one for normal and the others for four types of attack ”DoS, Probe, U2R, and R2L”. Figure (6) shows the system architecture of CPN, and FCPN to classify this dataset into 23 classes one for normal and 22 for subtype of attacks, node number of clustering layer and output layer is 23 nodes.

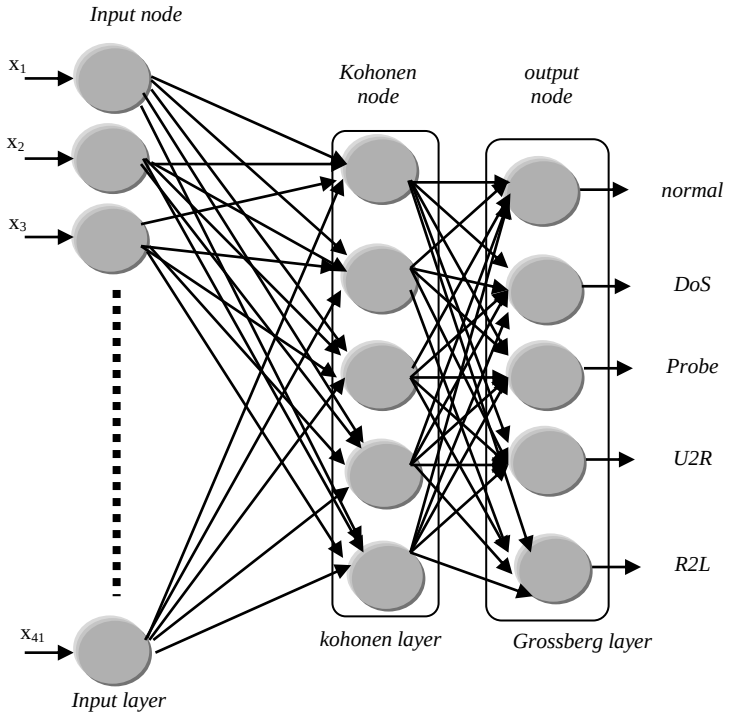

Figure 5. CPNN and FCPN Architecture for Multi Classification Network Intrusion (5 Classes)

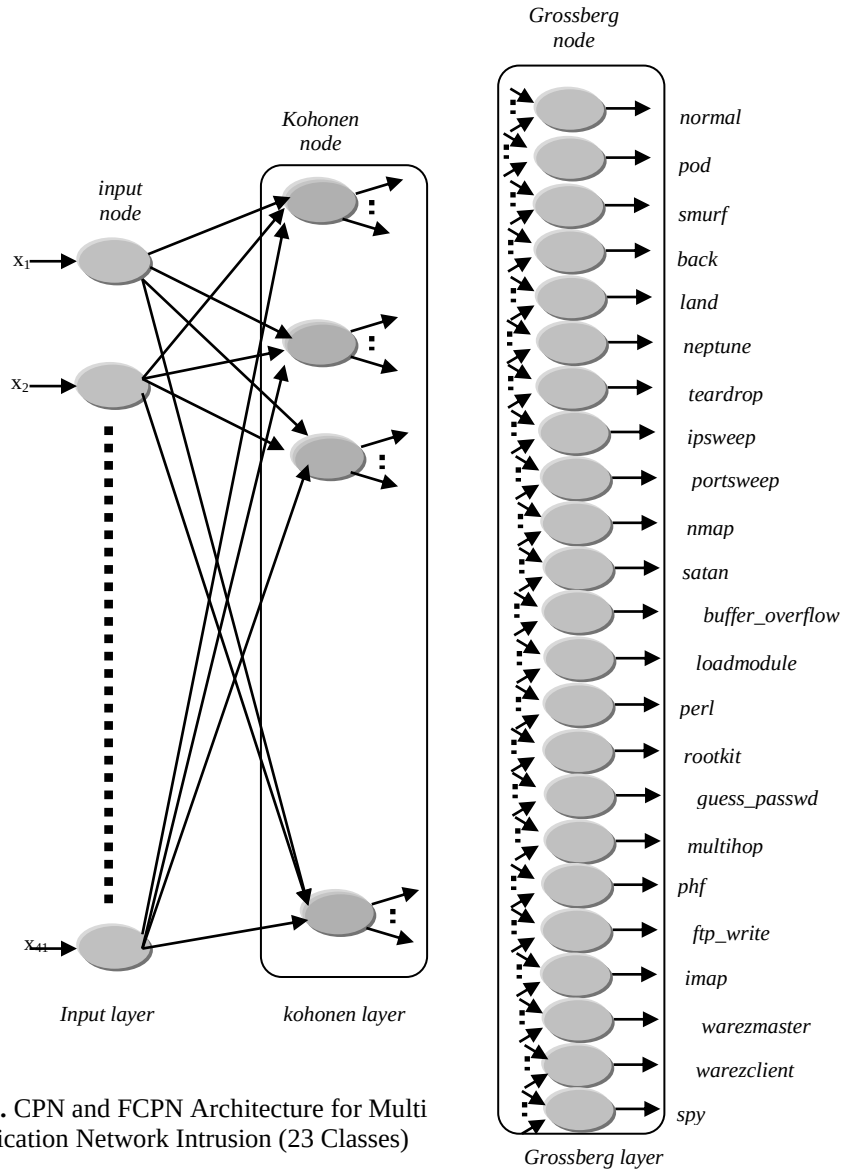

Figure 6. CPN and FCPN Architecture for Multi Classification Network Intrusion (23 Classes)

10. Experiments And Results

10. 1) Experiment 1

We applied fuzzy clustering algorithms (FCM, PCM, GK), (SFCM, SPCM, SGK) and CPN, and FCPNN on the 10%kdd dataset that contains (494020) records. In the first experiment, we applied these algorithms to classify this dataset into 23 classes or clusters. One for normal and the reset classes for the types of attacks { Dos (pod, land, back, neptune, teardrop and smurf), probe (ipsweep, portsweep, satan and nmap), U2R (buffer _overflow, loadmodule, perl and rootkit), R2L(ftp_write, guess_passwd, imap, multihop, phf, spy, Warezclient and warezmater)}. Table(3) shows the clustering results after training these fuzzy clustering algorithms, CPN, and FCPN. The results of classification rate obtained is 100%, but these fuzzy algorithms took different iterations and times.

Table (3). The Clustering Results after Training Fuzzy (FCM, GK,PCM), (SFCM, SGK, SPCM) algorithms, CPN, FCPN algorithms to classify dataset into 23 clusters

Amount	Sub type of attack	Samples rate
4	Phf	0.000810
107201	neptune	21.699729
3	Perl	0.000607
9	loadmodule	0.001822
1020	warezclient	0.206469
231	Nmap	0.046759
97277	Normal	19.690903
2203	Back	0.445933
8	ftp_write	0.001619
21	Land	0.004251
264	Pod	0.053439
280790	Smurf	56.837780
1247	ipsweep	0.252419
30	Buffer_overflow	0.006073
7	multihop	0.001417
2	Spy	0.000405
1589	Satan	0.321647
979	Tardrop	0.198170
20	warezmaster	0.004048
12	Imap	0.002429
1040	portsweep	0.210518
10	Rootkit	0.002024
53	guess_passwd	0.010728

Table (4) shows the result of the first experiment that using (FCM, PCM,GK), (SFCM,SPCM,GK), CPN, and FCPN clustering for 23 classes.
As shown in this table, SPCM was classified dataset faster than the other algorithms, because SPCM takes a number of iterations and time less than the other algorithms, but CPN takes time greater than the other algorithms.

Table (4). The Results of the (FCM, PCM,GK), (SFCM,SPCM,GK), CPN, and FCPN

Type of Clustering algorithms	Iteration number	Time second	Classification_rate
FCM	27	583.4	100%
GK	17	787.5	100%
PCM	14	307.9	100%
SFCM	9	195.2	100%
SGK	6	276.8	100%
SPCM	4	44.4	100%
CPN	10	1674.41	100%
FCPN	5	1222.74	100%

The "corrected KDD file" dataset that contains (311029) records were used in testing state on the fuzzy clustering algorithms (FCM, GK, PCM), and (SFCM, SGK, SPCM). Table (5) shows the comparisons between supervised(SFCM, SGK, SPCM)

and unsupervised (FCM, GK, PCM) fuzzy clustering algorithms for 23 classes with over all detection rate that obtained for FCM is equal (91.659) and for SFCM is equal (94.030), and detection rate that obtained for GK is equal (83.021) and for SGK is equal (92.672), and the detection rate that obtained for PCM is equal (94.284) and for SPCM is equal (95.971).

Table (5). Comparison between(FCM, GK, PCM), and (SFCM, SGK, SPCM) clustering algorithms

Performance measure	FCM	SFCM	GK	SGK	PCM	SPCM
Normal detection	34664	42090	33418	37814	60593	48863
Attack detection	250423	250371	224801	250423	237356	250430
Detection rate_normal	57.208	69.463	55.152	62.407	100	80.641
Detection rate_attack	99.995	99.974	89.764	99.995	94.777	99.998
False_alarm rate	42.792	30.537	44.848	37.593	0.0	19.359
Detection_rate	91.659	94.030	83.021	92.672	95.794	96.227
Times(Sec)	13.5	14	28.5	29.7	14.2 1	14.1

10. 2) Experiment 2

The same dataset (494020) records were used after preprocessing it in the training state to classify it into 5 classes, Table(6) shows the results of experiment for (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN.

Table (6). The clustering Results after Training (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN algorithms to classify dataset into 5 classes

Amount	Type of attack	Samples rate
97277	Normal	19.690903
391458	Dos	79.239302
52	U2R	0.10526
1126	R2L	0.227926
4107	Probe	0.831343

Table (7) shows the results after applying these fuzzy clustering algorithms (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN to classify dataset into 5 classes when fuzzification member value equals to (1.011). In this table, SPCM was classified dataset faster than the other algorithms, that's because SPCM takes number less of iterations and time than the other algorithms, but FCM takes times greater than the other algorithms. Classification rate that is obtained from all these algorithms is 100% in training stage.

Table (7). Results of the (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN clustering algorithms

Type of Clustering algorithms	Iteration number	Time second	Classification_rate
FCM	26	132.6	100%
GK	16	146.1	100%
PCM	12	66.36	100%

SFCM	9	43.9	100%
SGK	6	54.8	100%
SPCM	4	25.8	100%
CPN	10	1428.86	100%
FCPN	5	1164.05	100%

In testing state the ‘corrected kdd ‘ file that contains (311029) records are used in the fuzzy clustering algorithms (FCM, GK, PCM), (SFCM, SGK, SPCM), and CPN, FCPN algorithms. The comparisons between unsupervised and supervised fuzzy clustering algorithms (FCM, GK, PCM), (SFCM, SGK, SPCM), and CPN, FCPN for 5 classes with over all detection rate that obtained for FCM and SFCM is equal to (98.543), and detection rate that obtained for GK is equal to (80.836) and for SGK is equal to (81.155), and the detection rate that is obtained for PCM is equal to (99.955) and for SPCM and CPN is equal to (99.977), while FCPN got higher detection rate is equal to (100%) . Table (8) shows the comparison between (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN algorithms.

Table (8). Comparison between (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN algorithms

erformance measure	FCM	SFCM	GK	SGK	PCM	SPCM	CPN	FCPN
Normal detection	61948	61948	59683	60593	60523	60523	60593	60593
Attack detection	244548	244548	191741	191823	246200	24670	250366	250436
Detection rate_normal	97.813	97.813	98.498	100	99.884	99.884	100	100
Detection rate_attack	97.649	97.649	76.562	76.596	98.309	98.337	99.972	100
False_alarm rate	2.236	2.236	1.501	0.0	0.116	0.116	0.0	0.0
Detection_rate	98.543	98.543	80.836	81.155	99.955	99.977	99.977	100
Times second	2.7 sec	2.6 sec	5.8 sec	5.8 sec	2.6 sec	2.7 sec	330.831	329.053

10. 3) Experiment 3

The same dataset (494020) records were also used after preprocessing it in the training state to classify it into 2 classes, Table(9) shows the results of the experiment for (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN algorithms.

Table (9). The Clustering Result after Training (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN algorithms to classify dataset 2 cluster

Amount	Type of attack	Samples rate
396743	Attack	80.309097
97277	Normal	19.690903

While, table (10) shows the results after applying these fuzzy clustering algorithms (FCM, GK, PCM), (SFCM, SGK, SPCM), and CPN, FCPN to classify dataset into 2 classes. As shown in this table, SPCM algorithm was classified dataset faster than the other algorithms, because SPCM takes number of iterations and time less than the other algorithms, but CPN takes time greater than the other algorithms. Classification rate that is obtained from all these algorithms is 100%.

Table (10). Results of the (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN

Type of Clustering algorithms	Iteration number	Time second	Classification_rate
FCM	16	29.1	100%
GK	14	50.8	100%
PCM	12	22.2	100%
SFCM	9	17.0	100%
SGK	6	22.8	100%
SPCM	4	7.7	100%
CPN	10	1422.64	100%
FCPN	5	1166.78	100%

The ‘corrected kdd ‘ file that contains (311029) records were used in the testing state for fuzzy clustering algorithms (FCM, GK, PCM), (SFCM, SGK, SPCM), and CPN, FCPN, table (11) shows the testing results after applying these algorithms.

Table (11). The Results of testing state using (FCM, GK, PCM), (SFCM, SGK, SPCM), and CPN, FCPN algorithms

Type	Input	Output	DR
Normal	60593	60593	100
Attack	250436	250436	100

Table (12) shows the comparison between (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN algorithms in the testing state. This table shows SPCM is the faster algorithm, because it takes less time than the other algorithms.

Table (12). Comparison between (FCM, GK, PCM), (SFCM, SGK, SPCM), CPN, and FCPN algorithms in testing state

Performance measure	FCM	SFCM	GK	SGK	PCM	SPCM	CPN	FCPN
Times second	1.17	1.1	2.366	2.3	1.263	1.10	327.68	326.884

11. Conclusions

The main conclusions of this work are as follows:

1. Classification or accuracy improvement: the applied approaches based on unsupervised and supervised fuzzy clustering algorithms (FCM, GK, PCM, SFCM, SGK, SPCM), and CPN , and hybrid fuzzy with CPN that is called FCPN improved a high classification or accuracy rate.
2. Reduce training time: the intrusion detection mechanisms which are used took a few time for training dataset as compared to the other approaches.
3. Reduce computational overhead: the approaches which were used in this work reduce memory and computational overhead during the training and testing process. Because these approaches took less number of iterations and few time for execution.
4. Architectural framework improvement: the application of these approaches made the intrusion analysis engine more simple and efficient.
5. Detection improvement: these approaches obtained a high detection rate and low false alarm for KDD CUP 99 dataset. It has been found that FCPNN algorithm is the best approach.
6. IDS performance: To enhance the performance of IDS, this work proposes supervised methods such as (SGK, and SPCM), and also proposes FCPN method that satisfies the best performance.

REFERENCES

[1] Moradi M., Zulkernine M., 2003, "A Neural Network Based System for Intrusion Detection and Classification of Attack", Natural Science and Engineering Research Council Canada (NSERC).

[2] Abdolla M., 2009, "Enhanced Fast Attack Detection Technique For Network Intrusion Detection System", Ph.D. thesis, University Teknikal Malaysia Melaka.

[3] Toosi A., Kahani M., Monsefi R., 2006, "Network Intrusion Detection Based on Neuro-Fuzzy Classification", IEEE, international confernace on computing and information, Kualaumpur, Malaysi.

[4] Chimphlee W., Abdullah A., Sap M., Chimphlee S., Srinoy S., 2007, "A rough-fuzzy Hybrid Algorith for Computer Intrusion Detection ", the international Arab journal of information Technology, Vol.4, No.3.

[5] Chebrolu S., Abraham A., Thomas J., "Feature deduction and ensemble design of intrusion detection system ", www.elsevier.com/locate/cose . 2004.

[6] Panda M., Patra M., 2008, "some clustering algorithms to enhance the performance of the network intrusion detection system " ,journal of theoretical and applied information technology, pp.795-801.

[7] Skjolsvik S., 2007, "Framework for generating IDS benchmarking Datasets", MSC. Thesis, Department of Computer Science and Media Technology Gjovik University College, Norway.

[8] Tavallaee M., Bagheri E., Lu W., Ghorbani A., 2009, "A Detailed Analysis of the KDD CUP 99 Dataset", Proceeding of the 2009 IEEE Symposium on Computational intelligence in Security and Defense Application(CISDA).

[9] Chimphlee W., Abdullah A., Sap M., 2005, "Unsupervised Anomaly Detection with Unlabeled Data Using Clustering", Proceeding of the postgraduate Annual Research Seminar.

[10] Kendall K., 1999, "A Database of Computer Attacks for the Evaluation of Intrusion Detection System", thesis.

[11] Siripanwattana W., Srinoy S., 2008, "information security based on soft computing techniques ", Proceeding of the International Multi Conference of Engineers and Computer Scientists Vol I .

[12] Al-taey B. I., 2012, "Artificial Intelligent Techniques for Intrusion Detection and Classification", ph.D. thesis, College of Computer sciences and Mathematics, University of Mosul.

[13] Mukkamala S., Sung A., Abraham A., 2005, "Intrusion detection using an ensemble of intelligent paradigms", Journal of Network and Computer Application 167-182.

[14] Ching R., Cheng K., Hsieh C., 2009, "Using Rough Set And Support Vector Machine For Network Intrusion Detection", International Journal of Network Security & Its Applications (IJNSA), Vol. 1, No. 1.

[15] Gomez J., Dasgupta D., 2002, "Evolving Fuzzy Classification for Intrusion Detection", IEEE, Proceeding of the 2002 IEEE, Workshop on information Assurnace, United State Military Academy, pp 1-6.

[16] Ghorbani Y., Belacel N., 2003, "Y-Means: A Clustering Method For Intrusion Detection", 1CCECE 2003 CCGEI 2003, Montreal, IEEE, pp 001-004.

[17] ENGEN V., 2010, "Machine Learning For Network Based Intrusion Detection", Ph.D. thesis, Bournemouth University.

[18] Yan K., Wang S., Liu C., 2009, "A Hybrid Intrusion Detection System of Cluster-based Wireless Sensor Networks", International Multi Conference of Engineers and Computer Scientists, Vol. I

[19] Younis M., Khaleel B., 2011, "Enhance The Performance of Network Intrusion Detection System By Using Supervised And Unsupervised Fuzzy Clustering Algorithms", International scientific conference of Salahaddin University- Erbil (SU-ERBIL2011).

[20] Vlad Z., Ofelia M., Maria T., "fuzzy clustering in an intelligent Agent for diagnosis establishment ", inter-eng, 2009.

[21] Wang X., Garibaldi J. , 2005, "Simulated Annealing Fuzzy Clustering in Cancer Diagnosis", Automated Scheduling , Optimization and Planning (ASAP) Research Group.

[22] Zagoris K., Papamarkos N., Koustoudis I. , 2008, "Color Reduction using the Combination of the Kohonen self-organized feature map and the gustason-kessel fuzzy algorithm", Transaction on machine learning and data mining, Vol.1, No 1.

[23] Ali M. , Karmakar G., 2008, Dooley L., "fuzzy clustering for image segmentation using generic shape information ", Malaysian journal of computer science, Vol.21(2).

[24] Kumar A., Ghosh S., Dadhwal V., 2006, "A Comparison Of The Performnace Of Fuzzy Algorithm Versus Statistical Algorithm Based Sub-Pixel Classifier For Remote Sensing Data", proceeding part 7, pp 1-5.

[25] Kalyani S., Swarup K., 2010, "Supervised fuzzy c-means clustering technique for security assessment and classification in power systems", International Journal of Engineering, Science and Technology, Vol. 2, No. 3.

[26] Wasserman P. , 1989, "Neural Computing Theory and Practice", New York.

[27] Taylor B. J., 2006, "Methods and Procedures for the Verification and Validation of Artificial Neural Networks".

[28] Burks T. F., 2005, Shearer S. A., Heath J. R., Donohue K. D., "Evaluation of Neural-network Classification for weed Species Discrimination", Biosystems Engineering.

[29] Durai S., Saro E., 2006, "Image Mapping with Cumulative Distribution Function for Quick Convergence of Counterpropagation Neural Networks in Image Compression", World Academy of science, Engineering and Technology 16.

[30] Daniel Graupe, 2007, "Principles of Artificial Neura Networks", Advanced Series on Circuits and Systems-Vol. 6.