AL al-Bayt University

Prince Hussein Bin Abdullah

College for Information Technology

Computer Science Department

FLAME Clustering and Cuckoo Search Selection for Building a New Intrusion Detection Model

اللهب العنقودية و اختيار البحث الواقوق لبناء نموذج جديد لكشف التسلل

By

Kawthar Ahmad Alzboon

Supervisor Dr. Jehad Q. Al-Nihoud

Co-Supervisor Dr. Wafa' S. AL-Sharafat

2017

تفويض

انا كوثر احمد عقلة الزبون، أفوض جامعة آل البيت بتزويد نسخ من رسالتي للمكتبات او المؤسسات او الهيئات او الاشخاص عند طلبهم حسب التعليمات النافذة في جامعة.

التوقيع:

التاريخ:

اقراروالتزام بقوانين جامعة آل البيت وأنظمتها وتعليماتها

أنا الطالب: كوثر احمد عقلة الزبون **الرقم الجامعي :١٣٢٠٩٠١٠١٥**

التخصص: علم حاسوب **الكلية: تكنولوجيا المعلومات**

اعلن بانني قد النزمت بقوانين جامعة آل البيت وأنظمتها وتعليماتها السارية المفعول المتعلق باعداد رسائل الماجستير والدكتوراه عندما قمت شخصيا بأعداد رسالتي بعنوان:

FLAME Clustering and Cuckoo Search Selection for Building a New Intrusion Detection Model

وذلك بما ينسجم مع الأمانة العلمية المتعارف عليها في كتابة الرسائل والأطاريح العلمية. كما انني أعلن بان رسالتي هذة غير منقولة او مستلة من رسائل او أطاريح او كتب او ابحاث او أي منشورات علمية تم نشرها او تخزينها في اي وسيلة اعلامية، وتاسيسا على ما تقدم فأنيي اتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في جامعة آل البيت بالغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرج مني يعد صدورها دون ان يكن لي أي حق في التظلم او الأعتراض او الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

التوقيع: التاريخ:

FLAME Clustering and Cuckoo Search Selection for Building a New

Intrusion Detection Model

By:

Kawthar Ahmad Alzboon

Supervisor: Dr. Jehad Q. Al-Nihoud

Co-Supervisor: Wafa' S. AL-Sharafat

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master’s Degree in Computer Science

Deanship of Graduate Studies

Al al-Bayt University

2017

COMMITTEE DECISION

This Thesis (FLAME Clustering and Cuckoo Search Selection for Building a New

Intrusion Detection Model) was Successfully Defended and Approved on 22/5/2017.

<u>Examination committee</u>

<u>Signature</u>

Dr. Jehad Q. Al-Nihoud, (Supervisor)
Assoc. Prof. of Computer Science

Dr. Wafa' Al-Sharafat, (Co-Supervisor)
Assoc. Prof. of Computer Science

Dr. Omar Shatnawi, (Member)
Assoc. Prof. of Computer Science

Dr. Khaled Batiha, (Member)
Assoc. Prof. of Computer Science
Assoc. Prof. of Computer Science
(Hashemite University)

Dedication

To my parents.

The reason of what I am today.

There are no words that can express how grateful I am for your unconditional love, encouragement, and prayers of the day and night.

To my brother and sisters.

You all have been a great source of support and encouragement.

Thank you all.

ACKNOWLEDGEMENT

First of all, plentiful thanks to Allah the almighty the most merciful for enabling me to fulfill such achievement. I would like to express my deepest appreciation and gratitude to my main supervisor Dr. Jehad Odeh, for his deep knowledge, patience, kindness and encouragement. I am also indebted in preparation of this thesis to my co-supervisor Dr. Wafa' AL-Sharafatfor her support, guidance, persistent help and especially her patience during the preparation of this thesis.

I am also grateful to all the lecturers at the Computer Science Department at Al Al-Bayt University.

I would like to take a moment to express my deepest appreciation and gratitude to my father, who is a true inspiration for my life. I know I would never be where I am today without your guidance, support, endless love and your faith in me. I am so grateful that you are my father.

LIST OF CONTENTS

List of Figures

LIST OF TABLES

List of abbreviations

Abbreviation	Expression
IDS	Intrusion Detection system
LCS	Learning Classifier system
XCS	Extended classifier system
GA	Genetic Algorithm
CS	Cuckoo Search
RL	Reinforcement Learning
DoS	Denial of Service
DR	Detection Rate
FLAME	Fuzzy clustering by Local Approximation of Memberships
SVM	Support Vector Machine
ANN	Artificial Neural Network
KDD99	Knowledge Discovery and Data Mining
ACC	Accuracy
FPR	False Positive Rate
FAR	False Alarms Rate

Abstract

Recently, security has become an important challenge for entire networks. Networks have been hacked by unauthorized users by which their user data were accessed. Many methods have been applied to network security; such as firewalls, encryption, and antivirus. Intrusion detection system is one of these methods which monitors the network system and identifies the intrusions over the network.

This research is concerned with developing a model for intrusion detection systems. This model contains two phases; the first phase, focuses on feature filtration using FLAME algorithm, that has reduced the number of features space. The second phase, the extended classifier system (XCS), which can be implemented by providing an enhanced genetic algorithm operation. This enhancement is based on using a cuckoo search for selection in genetic algorithm along with different crossover and mutation probability instead of the traditional genetic algorithm.

The outcomes of the proposed system indicate that the performance of the developed model results with improved results compared to previous intrusion detection systems. The results show that an improvement of the detection and false alarm rate were achieved.

Chapter

Introduction

Introduction

Internet users are increasing over worldwide and giving the chance for attackers to increase their attacks to violate the systems resources and capabilities. Different methods have been applied and developed to discover illegal access, probing and any attack attempts that support end-users to reject or accept the incoming request (Zhao, 2007), one of these methods is firewalls and intrusion detection system that have been applied to prevent the traffic of unwanted incoming and outgoing traffic of data and intrusion detection systems (Rehman, 2003).

Intrusion detection system (IDS) focused on an intruder who can steal or change user system data (Raut, Singh, 2014). Anderson in 1980 suggested the idea of the IDS by designing a model that monitors the system. The model was based on the user behavior in detecting anomalies (Anderson, 1980).

Garden and others in 2014 stated that IDS are classified into data source or detection method:

1) Data Source

- Host-based intrusion detection system (HIDS): it receives information from an individual host and the operating system commonly traces and

- check records system. It is normally applied as an agent that is found on each host to be observed for the analysis of event logs, major system files, or the network looking for irregular changes or models for suspicious activities of traffic records (Yu Yingbing, 2012).

- Network-based intrusion-detection system (NIDS): it controls the traffic over the network packets and requests. So NIDS analyze network packets from those that appear unusual and flags them. It also can be applied over gathered data from various hosts for recognizing signs of infiltration (Yu Yingbing, 2012).

2) Detection Method

Network-based intrusion-detection system technique consists of two main categories:

- ❖ Misuse Intrusion Detection: which also known as the signature-based intrusion detection system, where it is based on the signatures of the attack, and it is established as generated alarms. These include attack signatures or pattern of certain movement or activity on the basis of normal activity known as intrusive (Hashem, 2013).

- ❖ Anomaly Intrusion Detection: which identifies a new type of intrusions and the infusion of normal use. Anomaly detection techniques are useful for unknown contra or new attacks due to the lack of previous knowledge about the fixed intrusions needed (Hashem, 2013).

Anomaly intrusion detection has been classified according to based detection techniques. One of these techniques is machine learning which can be classified into:

1. **1.** Neural networks: systems have learnt to search for next commands based on sequences of previous commands by a certain user. It suggests better solutions for problems of modeling user behavior in anomaly detection because they do not need any explicit user mode.
2. **2.** Fuzzy logic: this system is in charge of managing input parameters and input data invalidity.
3. **3.** Support vector machines: the system is a good generalized natural system with the capability to overcome the execration of dimensions (Yao, et al., 2006).
4. **4.** Learning classifier system: a set of rules (population of classifiers) guiding a performance in an unknown environment (Alsharafat Wafa, 2014).

NIDS has four major attack categories (Kumari, Shrivastava, 2012):

1. **1.** The Denial of Service (DoS): which is defined as "an attacker who makes some computing or memory resources too busy for authorized users to access". E.g: SYN flood, Smurf. (Shafi, 2006).
2. **2.** Remote to User (R2L): which is defined as " an attacker who send packets to a machine on a network, then exploit its ease of access to illegally gain local access as a user." E.g: Password guessing (Kumari, Shrivastava, 2012).

3. User to Root (U2R): which is defined as "the attack are corruptions in which hackers start the system with a normal user account and attempt to misuse vulnerabilities in the system to gain super user rights". E.g: Perl, xterm (Paliwal, Gupta, 2012).
4. Probe: which is defined as "Hosts and ports probes as predecessors to other attacks. Attacker scans the network in order to collect data or locate known vulnerabilities". E.g: Port scans, IP sweep (Shafi, 2006).

Motivation

Computer networks have been facing enormous security threats in which a new types of network attacks have appeared. A security techniques that are adaptable and flexible in order to protect computers from being hacked has become a global challenge. The intrusion detection system is an important technique to be applied in order to protect systems and networks against malicious activities. Anomaly based intrusion detection system is aimed to detect, prevent and report unauthorized activity in computer networks.

Problem Statement

Networks usage worldwide has undergone a large evolution, accompanied with an increasing number of hackers or attackers. Networks have a set of security needs, which concern about increasing the network reliability and availability, reducing the abuse and malicious attacks potentiality.

We need a system that has a high infiltration detection rate and it can deal with any new cases of attack. For previous systems, the DR is not high.

In order to have a safe and secure network environment; this study will show multiple artificial techniques to detect intrusions relying on an increased Detection Rate (DR) and decreased False Alarm Rate (FAR).

Main Contributions

Extended Classifier System (XCS) used a Genetic Algorithm (GA). GA is an evolutionary method that has a set of operations; selection, crossover and mutation. GA is used to generate new classifiers from existing classifiers.

The contribution of this research can be divide into:

1. Using the cuckoo search for selection operation in GA.
2. Using FLAME features filtration in XCS.

Structure of the Thesis

This study aims to develop a model of IDS in the extended classifier system. The rest of the thesis is categorized as follows:

Chapter 1: Introduction: In this chapter; introduction about a model, motivation, contributions, and problem statement.

Chapter 2: Research Background: In this chapter; the main concepts of the learning classifier system, XCS, GA, cuckoo search algorithm (CS), and Fuzzy clustering by Local Approximation of Memberships (FLAME) are illustrated.

Chapter 3: **Literature review**: in this chapter; we discuss the results of the comprehensive literature studies that formed this study.

Chapter 4: Methodology: this chapter presents the implantations of the proposed method.

Chapter 5: Experimental results and Evaluations: this chapter presents the evaluation of the result of the method implemented in this research and shows a comparison with results from some previous research.

Chapter 6: Conclusion and Future work: this chapter presents the final conclusion and the future work for the proposed work.

Chapter

Research Background

Introduction

This chapter introduces the environment of the XCS for performing detection engine and the FLAME algorithm for features filtration. In XCS, several algorithms will be used in the model such as; GA and a cuckoo search algorithm which will be presented in this chapter.

Learning Classifier System

Learning Classifier System (LCS) is considered a machine learning system, which was introduced by Holland in 1976 (Alsharafat, 2013). LCS includes rules that are called classifiers. A classifier system could learn and classify messages from the environment into general sets which depends on the type of LCS (Richards, 1995).

LCS has three major parts as follows (Bensefi):

1. Rule base: that signifies an adaptive, reactive, and evolutionary knowledge base for the LCS
2. Reinforcement Learning (RL): that manage of this knowledge's, adaptability base

3. GA: that manages the evolution.

LCS is applied to handle problems of interaction with the environment by detecting intrusion attempts (Shafi et al, 2007). Intrusion detection has been detected activities that break the security policy of networks (Kumari, Shrivastava, 2012).

LCS has three major categories:

1) Strength based LCS, which is called Zeroth Classifier Systems (ZCS). The ZCS was proposed by Wilson in 1994 (Sigaud, Wilson, 2007), which it has a condition and an action part where each a classifier comprises one evaluation variable that includes its accumulated reward estimation brought by its firing and fitness for the process of population evolution (Sigaud, Wilson, 2007). Figure (2.1) shows the ZCS work flow:

1- Initialization

- Initialize population of classifiers R = {R1 ,R2 , ...,RN}

2- Performance

- Receive a binary encoded input.
- Determine an appropriate response based on the rules whose condition matches the input .
- Produce a classification decision and update rules' fitness values.

3- Reinforcement

In case of successful classification, apportion a scalar reward R to the system classifiers according to a reinforcement scheme .

4- Discovery

Change one individual of the classifier population by applying GA.

Figure (2.1): Work Flow ZCS (Tzima, et al., 2009)

2) Anticipation based LCS, which is called Anticipatory learning classifier systems (ALCS). The ALCS rules comprise three parts; condition, action and effect. The accuracy of prediction effects depends on a particular action under a particular condition. The ALCS is concerned with what will happen after executing an action (Alsharafat, 2013).

3) Accuracy based LCS, which is called Extended Classifier System (XCS). In XCS, rule predicts a particular reward and has a particular fitness. It retains rules that predict lower rewards

as long as those predictions are accurate. In this thesis, we will shed the light on the extended classifier system (XCS) because it widely used in different applications as in IDS.

Extended Classifier System (XCS)

XCS is the most popular LCS and it is widely used in different applications as in IDS (Bernad´oMansilla, Garrell-Guiu, 2003). XCS was introduced by Wilson in 1995, which classified as a rule based system where rules populations are called classifiers (Shafi et al., 2006). Each rule contains two parts: condition and action.; condition part (“the body of the rule”) (Dam, Abbass, 2008) is represented in binary system as :{ 0,1, #}; the symbol # means don't care. The action part (“the prediction of the classifier”) (Dam, et al., 2008) is presented as (0,1) (Bernad´oMansilla, Garrell-Guiu, 2003).

XCS is based on two factors: RL and GA. RL (or credit assignment) which allocates the incoming reward from the environment of the classifier which is responsible for the reward received (Holmes, et.al, 2002). RL is designated to know how the classifier will be useful in the future reward and to feed the development of better rules (Lanzi, 2008). GA discovers the search space by generating a new rule into the system (Dam et al, 2005). Once a new rule is generated; the population gets scanned to examine if the new classifier already exists or not. So, if a new classifier not a duplicate rule than the new rule will be added to the population. Also, the number of existing is incremented by one. The main components of XCS are shown in figure 2.2.

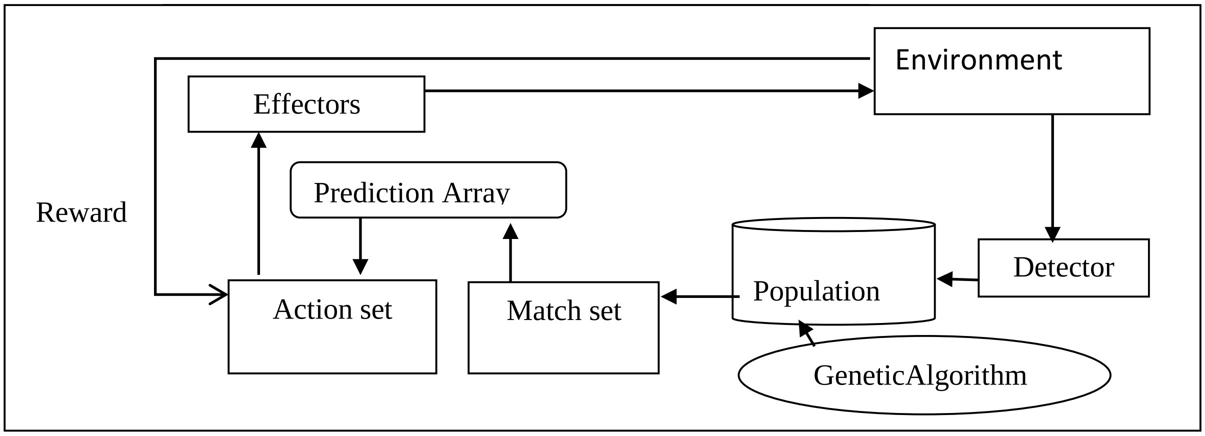


Figure (2.2): Components of XCS (Alsharafat, W. , 2013)

1. **Detector:** receives inputs from the environment message that represents the network traffic feature. These features are classified into important features that play a major role to detect attacks. In opposition, irrelevant features replaced by #. Features consist of a condition part that is utilized to detect network attacks.
2. **Match set [M]:** a set of classifiers in which conditions part must match the condition part of input features of the environment.
3. **Prediction Array:** this is formed for each action in [M] depending on its fitness-weighted average of the prediction of rules in each [A] (Bull, kovacs, 2005).
4. **Action set [A]:** a set of classifiers in [M] that support the action chosen.
5. **ffector:** it fires the rule action to the environment. The result can be normal, Probe, U2R, R2L or DOS. Figure 2.3 shows the pseudo code XCS (Butz., et.al, 2007):

- Initialization Population of rules .
- Match set formed in response from environment
- Action selected from match set.
  - Highest fitness
- Rules advocating the same action form the action set
- Receive a reward r from the environment for executing the specified action
- Update the predicted reward for each rule in the action set
  - $p \leftarrow p + \beta\ (r-p)$
- Update the predicted error for each rule in the action set
  - $\varepsilon \leftarrow \varepsilon + \beta\ (|r-p| - \varepsilon)$
  - β = estimation rate
- If $\varepsilon < \varepsilon_0$, set prediction accuracy k=1
- Otherwise, set prediction accuracy
  - $k = \alpha(\varepsilon_0/\varepsilon)^v$ for some α,v>0
- Calculate relative prediction accuracy
  - k' = k(rule) / (sum of k for all rules in action set)
- Update the fitness of each rule
  - $f \leftarrow f + \beta\ (k' - f)$
  - α = learning rate
  - β = estimation rate

Figure (2.3): Pseudo code XCS (Butz., et al., 2007)

More ever, each classifier keeps certain additional parameters (Butzi, Wilson, 2001):

- The prediction error (ε): estimates the errors made in the predictions.
- The prediction p: estimates (keeps an average of) the payoff expected if the classifier matches and its action is taken by the system.
- The fitness f: denotes the classier's fitness.
- β: is the learning rate for p; ε; f.
- a, ε, and v: are used in calculating the fitness of a classifier.
- k: prediction accuracy.

GA is an empirical search algorithm that depends on ideas of natural selection. GA is employed in intelligent exploitation of a random search with a defined search space in order to solve a problem (Goldberg, Holland, 1988).

GA is based on the idea of the existence of the fittest, where a population is produced to create innovative search strategies. Initially, GA consists of a set of individuals called population, which is used to represent possible solutions for the specified problem. Then by performing selection methods, crossover and mutation; GA will iteratively create a new individual from the old population (called a generation) (Mitchell, 1999).

GA is used to solve complex problems in various fields. A GA uses genetic concepts to encode problems into a generation (a group of individuals), then it simulates the generation evolution by applying mathematic genetic operators (selection, crossover and mutation) to define the best solution (individual) over a finite number of generations. The definition of “best” is accompanied with a fitness function that describes a given individual and decides if it is better or worse than other individuals. These steps are then repeated until a termination condition is fulfilled (Mitchell, 1999).

GA steps will be presented in more details in the next section. General (Simple) GA is illustrated in Figure 2.4.

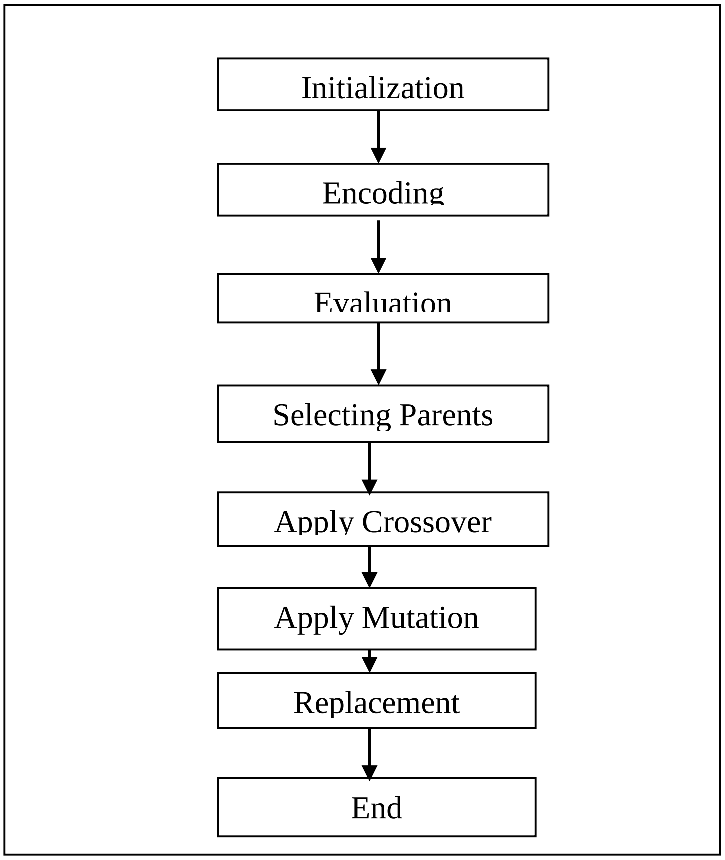


Figure (2.4): GA flow chart(Mitchell, 1999)

2.4.1 GA elements

GA comprises a set of elements:

1. **Population**

The basic terms of the population of individuals are gene and chromosome. A population is a group of individuals of a definite size. Individuals of a population are sets of task parameters coded in the form of chromosomes, that means solutions; or else, they are called search space points. Individuals are generated in a random manner to represent initial values for present generation which represents an initial search space for GA.

2. Evaluation

The fitness function is adopted to evaluate the chromosome fitness in which the fitness value shows the quality of each chromosome (Alabsi, Naoum, 2012).

3. Encoding

Encoding is one of the most important methods in GA in order to represent suitable solutions. Choosing the right representation method improves the effectiveness of GA in solving problems, there are different ways for encoding; such as binary, real or integer (Mitchell, 1999).

4. Selection

Selection is the process of choosing individuals from existing populations as parents, in order to implement crossover and mutation on them to reproduce new individuals (offspring or child). Then, the decision to be taken is how can individuals be selected from a current generation. By applying Darwin principle "survival of the fittest"; individuals with highest fitness values will take the advantage to live for a long time and cross over with low fitness individuals.

Different selection methods are used in this scope: roulette wheel selection (RWS) and ranking selection. The RWS is a selection method that selects high probability parents with high fitness values. While, ranking selection is an alternative method which prevents fast convergence and slow finishing problems. There are different ways to perform this method, but the simplest of all is a linear ranking method (Alabsi, Naoum, 2012).

5. Replacement

Comparison between several chromosomes is conducted in order to choose the best. Replacements include; binary tournament and triple tournament. Binary tournament takes two chromosomes and depending on their fitness function, it chooses the best one and ignores the second. Triple Tournament replaces the worst two among three chromosomes by considering the chromosome with the highest fitness value (Alabsi, Naoum, 2012).

2.4.2 GA operator

GA comprises two main types of operators to be used to reproduce new individuals in next generations. These operators are crossover and mutation.

1. Crossover

Crossover is one of the main characteristics distinguishing GA from another revolution techniques. A crossover is the process of genes exchange between two individuals (chromosomes) to reproduce new individuals that inherent their parent's behavior(Goldberg, Holland, 1988). The crossover was used at first to represent the search in the parameter space. Then it is concerned with finding a way to keep the information stored by parent chromosomes as long as possible as they are considered to be good chromosomes that have resulted from process selection (Mitchell, 1999),(Goldberg, Holland,1988). Crossover decision implementation, for all genes, depends on the value of crossover probability (*P*c).

Crossover probability is how often a crossover is performed if there is no crossover, in which, the offspring is in an exact copy of its parents, and then the offspring is made by applying crossover. Crossover methods determination is dependent on encoding method and problem type (Nadi, Khader, 2011).

$$p_c = \begin{cases} k_1(f_{max} - f')/(f_{max} - \bar{f}) & \text{if } f' \geq \bar{f} \\ k_3 & \text{otherwise} \end{cases} \quad \text{....... (2.1)}$$

Where: k*1, k3* ≤ 1.0 are constants, *f* is the fitness of the individual, *fmax* is best existing fitness, *f'* is the largest fitness of the parents that are selected for crossover, and $\bar{f}$ is the average fitness of the population (Nadi, Khader, 2011).

Various crossover strategies are present:

a. Single crossover position

This is the simplest method for crossover. Crossover determination can be accomplished by using crossover probability and selecting random position k, where; k ∈ {1,2,.., *l*-1}, *l*= is chromosome length) to produce two Children (chromosomes) by intersecting parents' chromosomes at k position (Mitchell, 1999).

b. Two-point crossover

This strategy executes better than single point crossover or to some extent they are considered to be equivalent. Crossover can be accomplished by using crossover probability and selecting random position k_1, k_2 (k_1, $k_2 \in \{1,2,..,l\text{-}1\}$, $k_1<k_2$) where the genes between k_1 and k_2 are switched (Goldberg, et al., 1988).

c. Uniform crossover

Uniform crossover varies from other strategies, it is as genes are randomly exchanged by using probability.

2. Mutation

Random changing in genes in individual chromosome is applied; to avoid local maxima and produce new individuals that differ from the existing for more exploration of the search space. The decision for implementing mutation, for all genes, depends on the value of mutation probability (*Pm*).

Mutation probability can be defined as how frequent will a part of a chromosome mutate(Nadi, Khader, 2011).If no mutation is achieved; the offspring is then considered after the crossover without any changes. If mutation is 100% in which the whole chromosome is changed (Goldberg, Holland, 1988); the below equation of mutation probability (*pm*) is appli

$$p_m = \begin{cases} k_2(f_{max} - f)/(f_{max} - \bar{f}) & \text{if } f \geq \bar{f} \\ k_4 & \text{otherwise} \end{cases} \quad \text{..... (2.2)}$$

Where k*2,* k*4* ≤ 1.0 are constants, *f* is the fitness of the individual, *fmax* is best existing fitness, *f'* is the largest fitness of the parents that are selected for crossover, and $\bar{f}$ is the average fitness of the population (Nadi, Khader, 2011).

Cuckoo Search

Metaheuristic algorithms are inspired by natural phenomena as such as; partial swarm optimization (PSO) which was inspired by fish and swarm intelligence and cuckoo search has brood parasitism behavior of the cuckoo birds. The major two properties of metaheuristic algorithms are selection of fitness, which depends on searching around for the present solution and selecting the best, and adaptation to the environment, in which the algorithm explores the search space.

Cuckoo search (CS) is an evolutionary optimized algorithm that was first presented by Xin-She Yang and Suash Deb in 2009 (Yang, Deb, 2009). Cuckoo search has a brood parasitism behavior of the cuckoo species with the Lévy flight behavior. These birds lay aside their eggs in a host nest and imitate external properties of host eggs such as color. In this strategy is possible for the host to throw away the cuckoo's egg or leave its nest to build a new one in another place (Moghadasian, Hosseini, 2014). CS has two types of behavior; cuckoo breeding behavior and Lévy flight behavior. The next sections detail these types.

Cuckoo Search Behavior

2.5.1.1 Cuckoo Breeding Behavior

The generation process of the cuckoo search algorithm depends on three rules:

1. Each cuckoo selects a random nest where it lays one egg at a time.
2. High quality egg nests will proceed to the next generation.

3. A fixed number of host nests are available, where, each host is able to detect a strange egg with a probability *p*a [0,1], and also, the host bird can discard the egg or leave the nest to construct a new nest in another place. (Kanagaraj, et al., 2013).

2.5.1.2 Lévy flight Behavior

The Levy flight considers a random walk, that depending on the step size. The step size is mainly subject to heavy-tailed probability distribution (Valian, et al., 2011), (Roy, Chaudhuri, 2013). Levy flight was introduced by Benoit Mandelbrot by applying a special definition for the distribution of step sizes. Levy flight is utilized to designate a separate network rather of a continuous space (Roy, Chaudhuri, 2013). Cuckoo Search and Lévy flight (CS) were reviewed according to the pseudocode shown in

```
Cuckoo Search via Lévy Flights
begin
  Objective function f(x),  x = (x1, ..., xd)^T
  Generate initial population of
      n host nests xi (i = 1, 2, ..., n)
  while (t <MaxGeneration) or (stop criterion)
    Get a cuckoo randomly by Lévy flights
      evaluate its quality/fitness Fi
    Choose a nest among n (say, j) randomly
    if  (Fi > Fj),
        replace j by the new solution;
    end
    A fraction (pa) of worse nests
        are abandoned and new ones are built;
    Keep the best solutions
        (or nests with quality solutions);
    Rank the solutions and find the current best
  end while
  Postprocess results and visualization
```

Figure (2.5): Pseudo code Cuckoo Search (Yang, Deb, 2009)

A new solution is denoted by $x^{(t+1)}$ and a cuckoo is represented by i, then a Levy Flight is performed as in equation (2.3) (Guerrero,et al., 2015):

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus L\acute{e}vy(\lambda) \qquad ....(2.3)$$

Where: current solution is denoted by $X_i^{(t)}$, a step size is represented by α; it should be relevant to the scales of the problem of interest, where *a* ≥ 0 and the product $\oplus$ means an entry-wise multiplications (Guerrero, et al., 2015). Levy flight is a random walk while the random step length is drawn from a Levy distribution (Valian, 2011).

$$L\acute{e}vy \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \qquad .... 2.4$$

Where *t* is The number of a current generation (Time), and λ is a constant between 1 and 3 (Guerrero, 2015).

Levy distribution possesses an infinite variety with an infinite mean along with a power-law step size of a heavy tail (Roy, Chaudhuri, 2013). Levy walk generates some new solutions around the best reached solution so far, which will speed up local search (Guerrero, 2015). But, a substantial fraction of new solutions should be produced by the far field random distribution, whose locations must be far enough from the current best solution; this will in turn guarantee that the system will not be trapped within a local optimum (a solution that is optimal (either maximal or minimal) within a neighboring set of candidate solutions(Kosheleva,Kreinovich, 2016)), (Yang, Deb, 2009).

FLAME Clustering

The main idea of clustering data is to reduce the amount of data by categorizing or grouping similar data items together (Barbakh, et al., 2009). There are different methods to be used for data clustering. Fuzzy Clustering by Local Approximation of Memberships (FLAME) is one of the known clustering algorithms**;** it identifies clusters according to the dense portion of the dataset. FLAME is based on the neighborhood relationships between objects that were applied to force neighboring objects memberships in fuzzy membership space (Sampath, Prabhavathy, 2015). FLAME data clustering algorithm runs through three steps (Fu, Medico, 2007):

Step 1: Extraction of structure information

(a) Create a neighborhood graph to connect each object to its K-Nearest Neighbors (KNN);

(b) Estimate a density for each object based on its proximities to its KNN;

(c) Objects are classified into 3 types:

1. Cluster supporting object (CSO): an object with higher density than all its neighbors.
2. Cluster outliers: an object with lower density than all its neighbors, and even lower than a predefined threshold.
3. The rest.

Step 2: Assigned fuzzy membership by local approximation

(a) Initialization of fuzzy membership:

1. Each CSO is assigned with fixed and full membership to itself to represent one cluster.
2. All outliers are assigned with fixed and full memberships to the outlier group.
3. The rest are assigned with equal memberships to all clusters and the outlier group.

(b) Then the fuzzy memberships of all of the 3 types of objects are updated by a converging iterative procedure called Local/Neighborhood Approximation of Fuzzy Memberships; in which the fuzzy membership of each object is updated by a linear combination of the fuzzy memberships of its nearest neighbors.

❖ The weights defining how much each neighbor, will contribute to an approximation of the fuzzy membership of that neighbor is calculated as *waxy* with (Fu, Medico, 2007).

$$\sum_{y \in N(x)} Wxy = 1 \qquad \ldots.(2.5)$$

❖ Local/Neighborhood Approximation Error (LAE/NAE), which is defined as the following(Fu, Medico, 2007):

$$\mathrm{E}\,(\{\mathrm{p}\}) = \sum_{x \in X} ||p(x) - \sum_{y \in N(x)} -W_{xy}\, \mathrm{P(y)}||^{2} \qquad \ldots\ldots(2.6)$$

❖ In FLAME, Eq (2.5) is minimized to calculate a set of memberships vectors under some constraints (in addition to the natural constraints on fuzzy membership vectors) derived in the first step, that is, fixing membership vectors of CSOs and outliers to avoid the trivial solutions where all p(x) are the same.

❖ The NAE can be lessened by solving the following linear equation with a unique solution that is the unique global minimum of NAE with a zero value:

$$P_k(x) - \sum_{y \in N(x)} - W_{xy} Pk(y) = 0\ ,\ \forall x \in X, k = 1, \ldots, M \qquad \ldots.(2.7)$$

❖ In which M is the number of CSOs plus one (for the outer group). Following the iterative procedure can be applied to solve these linear equations:

$$p^{t+1}(x) = \sum_{y \in N(x)} - Wxy\, p^t(y) \qquad \ldots(2.8)$$

Step3: Construction of cluster with the fuzzy memberships

(a) One-to-one object-cluster: select each object to the cluster that has the highest membership.

(b) One-to-multiple object-clusters: select each object to the cluster that has a membership higher than a threshold

Chapter
Literature review

Introduction

This chapter presents some previous research on intrusion detection systems and classifiers techniques applied in the IDS data. It presents algorithms used by IDS to reduce features space.

Anomaly Intrusion Detection System

Jyothsna and Prasad (2011) focused on operational architectures and several techniques in the anomaly intrusion detection system. Their classification depended on the behavior of the system. Among these techniques; statistical models, cognition models, machine learning based detection techniques, kernel based online anomaly detection, detection models that are based on computer immunology and models based on user intention were implemented. They presented major features of several intrusion detection systems platforms that are currently available.

Raut and Singh (2014) conducted a study concerned with an anomaly based intrusion detection system (ABIDS) and techniques. They showed a detailed several techniques of ABIDS, that are, statistical anomaly detection, data-mining, knowledge based and machine learning. Statistical anomaly is used statistical properties and it is classified into two types: operational model and marker model. Data-mining detected known attacks and it is classified into clustering and classification. Knowledge based detection collected knowledge about specific attacks and weakness in the system and then applied this knowledge to exploit the weaknesses of the attack and to generate alarms; it is classified as state transition analysis, expert system and signature analysis. The machine learning is based on learning system and performance improvement over time; it includes three categories: Neural Networks, Fuzzy Logic Approach and Support vector machines.

Intrusion Detection System With XCS

Shafi and Abbass (2006) evaluated XCS based on three principles; quantified performance, which is a fraction of the cases that are classified correctly and calculated through a window to exploit the trials (typically 50), the ratio optimal amount of population size, and the upper limited of exploitation paths. This research examined the issue of the standard three early stops on a subset of benchmark intrusion detection KDD99 data. They concluded that the smaller size of the population the better the accuracy and the less the computational cost because when they increased the size of the population it did not increase the accuracy. Also, they reduced the number of features from 41 to 29. The population of a size higher than 2000 came with an accuracy of 95%.

Alsharafat (2010) developed a model for intrusion detection system that comprises two phases; during the first-phase; a process to filter features is generated in order to select the best set of features for each implemented type of network attacks by using artificial neural network (ANN). The second phase includes designing an ID by using an extended classifier system (XCS) with an internal classifier modification generator to obtain better detection rates (DR). The proposed model System can successfully and professionally detect attacks. R2L and U2R were spotted at low rate. This can denote the small number of records given to these attacks in KDD'99 dataset and test set. The detection rate for the ANN-XCS model was 98.01% with false-positive rate of 0. 9%.

A new method for intrusion detection system based on data mining and improved XCS was presented by **Panahi (2013)**; snort software was used with network intrusion detection system (NIDS). It included a named package listing as package record and other features. Numerous software is used for data mining in different fields. The fitness of the rules in XCS improves each rule for survival and participation in the production process associated with how it answers the training data. The results showed that the enhancement of XCS revealed a better detection rate that arrived to 94.83%, while snort got detection rate that arrived to 70.27%.

Yazdani and others (2013) improved the extended classifier system algorithm by using a new method. XCS was used to identify attacks on the databases. XCS was prepared and trained by using a set of existing examples and used reinforcement learning techniques to identify attempts conducted to intrude the databases and provided preventive incentives against them. The detection rate arrived at 91% for various types of known attacks to the databases.

Intrusion Detection system with Support Vector Machine, Neural Network

Shrivastava and Jain (2011) proposed a model for improving anomaly intrusion detection in order to gain a high detection level and a low false positive value based on using rough set which reduces features data set and SVM to test and train the data. The proposed model used only 6 out of 41 features. The model decreases CPU and memory utilization for the system and it is trusted in detecting intrusion. The accuracy of the proposed system is 95.98 % and a false-positive rate of 7.52%.

Tiwari (2013) generated a hybrid model for intrusion detection systems by using a firefly algorithm (FA) for feature selection a radial basis function (RBF) Neural Network which is a kind of three-layer feed-forward neural network and a rough set theory which is a mean to deal with intelligent data analysis and data mining. The KDD99 dataset was used comprising 32 features and four types of attacks with DoS attack detection rate of 0.99, Probe detection rate of 0.98, R2L detection rate of 0.97 and U2R detection rate of 0.95.

Intrusion Detection System By Using Genetic Algorithm

Agravat and Rao (2011) used fuzzy Genetic-based Learning algorithms to detect intrusion in the network. They minimized fuzzy rules number and maximized classification rate. They used fuzzy Genetic Algorithm for Misuse Detection to be calculated and tested over KDD 99 dataset. 20 features along with the 41 from KDD Cup 99 were used. In addition to that, they also used the parameters; A number of elite solutions = 20 %, crossover probability = 0.9, mutation probability = 0.1 and the number of generations = 50. The outcomes were of Precision = 0.9979, Recall = 1 and Accuracy = 0.985.

Kadam and Jadhav (2013) proposed a model for intrusion detection systems by using genetic algorithm as a model to produce rules for different types of inconsistent connections for intrusion detection system for improved accuracy and detection rate. The outcome showed that they used nine features, by using in the GA, Crossover probability = 0.8, mutation probability =0. 08. Detection rates normal attack = 81.25%, DOS attack = 97.80%, Probe attack = 76.12%, R2L attack = 23.00%, U2R attack = 30.70%, and detection rate arrived to 91.025%.

Pawar and Bichkar (2014) implemented a genetic algorithm for the intrusion detection rule generation that included different variables like; population size, selection, crossover and mutation along with six features. They concluded that the detection accuracy increases in intrusion detection system along with the population size increase. The results showed that, when using roulette wheel selection, two-point crossover with a crossover rate of 0.6 and uniform mutation with mutation rate of 0.01, highest detection accuracy of intrusion detection system can be detected with a value of 98%.

Danane and Parvat (2015) proposed a model by using a fuzzy algorithm and genetic algorithm for intrusion detection system. They aimed to achieve system improved accuracy, memory allocation and execution time for intrusion detection systems. The results showed that by using six feature and crossover probability of =0.8, mutation probability =0.088 and accuracy=0.98; KDD99 dataset is a point of reference dataset to implement a model.

Patel and Buddhadev (2015) implemented a method for predicting rule detection by using genetic algorithm (GA) to generate a rule base for intrusion detection systems. The method included two stages; using KDD Cup 99 dataset to generate the rule base in order to train the parameters (number of generations, mutation rate, and the probability of crossover). Then, it tests the system using this rule base and the KDD Cup 99 testing dataset. They are using one-point crossover and the probability of crossover = 0.7, mutation rate= 0.01 and detection rate =98.7%.

Sasan and Sharma (2016) developed a hybrid model for IDS, where the model analyzed the behavior of network data depended on prior features and used the machine learning techniques with misuse detection. In the proposed model used 29 features with accuracy rate= 88.23%.

Chapter
Methodology

Introduction

This chapter presents how to use our system and how to obtain the results. It displays the proposed model and explains the model details.

This research explains the improvement of the XSC for IDS and assess system performance by calculating the system DR and FAR.

A method for feature filtration by using FLAME and a modification of GA operation to reach optimal or most near optimal solutions will be presented in this study.

The proposed research mainly consists of two phases; selecting features using the FLAME algorithm in order to decrease the number of features, since some of the features are irrelevant and redundant, which results lengthy detection process and degrades the performance of an intrusion detection system (IDS) (MukherjeeSharma, 2012).

Using the CS in the selection; the genetic algorithm operates in an extended classifier system. In system evaluation level, two values are to be calculated; the DR and FAR. A strong system must possess high DR and low FAR. In this chapter, all phases will be discussed in details to explain the suggested enhancement.

Feature Filtration

The KDD'99 dataset comprises a set of 41 features each feature that is coming from a connection and a label that specifies the connection records' status as normal or specific attack types. KDD'99 dataset is separated into training and testing data sets. The proposed system uses 10% of KDD'99 dataset because it does not occur Java heap. Training data are considered to be a condition part that grasps feature values and also as an action feature that holds the attack label.

FLAME is implemented to reduce features number of the dataset from a training dataset and select the best features by applying FLAME which is mention explained in chapter 3. FLAME clusters contain three main clusters:

1. Inner: an object with density higher than its neighbors.
2. Outer: an object with density lower than its neighbors and lower than a predefined threshold.
3. Rest: an object with a density that lies between the inner and the outer object and close to the threshold.

Proposed System Model

KDD 99 data set was used as an environment in this research**.** The following figure (4.1) displays component of the proposed system model XCS.

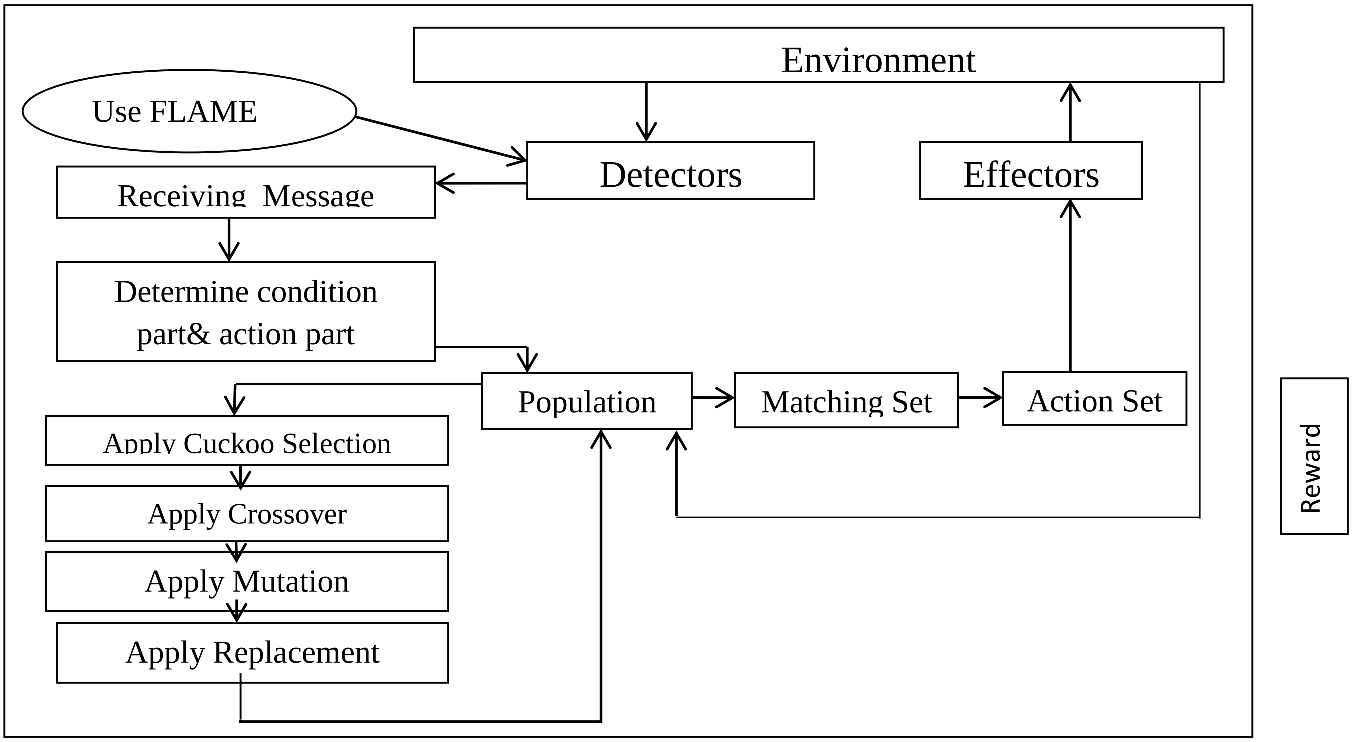


Figure 4.1:Proposed System Model XCS

Environment

The KDD'99 data set was used as an environment in this research; it contains 5 million records while each record comprises 41 features. KDD'99 divided it into two subsets; training dataset in which the system receives data and process and testing dataset that evaluates the system. In this research, we used a subset of training and test data set by applying 10% KDD'99 using 500000 records as training and 300000 records testing.

Detectors

The detector classified the data by receiving a message from an environment, then it represented the message in real representation. Every massage composed into the condition and the action. Also, detectors filters the message from repetition and noise to obtain the most significant features by using FLAME.

Population

GA was used to generate new classifiers from existing classifiers. In this research, cuckoo search algorithm was implemented in the selection operator and dynamic probability for crossover and mutation for the GA parameters to produce the best generation of rule classifiers from existing classifiers. Host nests were represented as a population and each cuckoo egg was represented as a solution. CS is explained further in chapter 3.

The resulted rules after using CS selection in GA will be stored in rules pool in order to be used in the next step that examines the dataset testing. Here, CS was used for selection to achieve better results and using uniform crossover and random mutations. The following figure (4.2) displays the Pseudocode for GA and CS.

Start a New Generation:
1): Determine a population size.
2): Represent data using real representations.
For each population in the rule pool, do:
3): choose the chromosome by using cuckoo search for selection.
3.1): Generate a new set of solutions (host nests) but keep the
    Current best (say, *i*) randomly by Lévy flights
    incorporating with inertia weight, *w*, which controls the search ability
    Evaluate new solution fitness *Fi ;*
    Get a selected set of host nests among n (say, *j*) and
    calculate its fitness *Fj* ,
    if (F*i*> F*j*)
    Replace j by the new set of solutions, *i*;
    End
    A dynamic fraction probability, P_a of worse nests is
    abandoned and a new nest (set of solution) is built;
    Keep the best nests with quality solutions;
    Let the best nests become as initial chromosomes;
    Evaluate each individual's fitness;
    Select pairs to the best ranked individuals;
4): Apply crossover
5): Apply mutation
6): Save the created rules
7): Go to the next population

Figure 4.2: Pseudocode Algorithm of Genetic Algorithm and Cuckoo Search

Figure: 4.2 Pseudocode for Genetic Algorithm and Cuckoo Search

Matching Set

At this stage of the proposed model; the researcher will try to match the condition part of classifiers received from an environment with the data identification existing rules.

Action Set

The set of classifiers in the matching set calls the action that is actually chosen. In the proposed work there are four actions where each action depends on the type of attack (DOS, Probe, R2L, U2R) and detect intrusions to be alerted.

Effectors

Firing the rule action to the environment; expected result can be normal, Probe, U2R, R2L or DOS.

Chapter

Experimental results and Evaluation

Introduction

This chapter focuses on introducing experimental results and the evaluation of the proposed work. Also, a comparative assessment with several researchers who presented experimental results that focus on IDS in a network environment is also presented.

This thesis presents the performance before reducing the 41 features and after the features were filtered. For performing experiments we used desktop computer Acer with core i7, 6.00 GB of RAM and hard disk 500GB, under windows 7 platform and eclipse LUNA for implementing JAVA.

Performance Measurements

In order to address the comparative assessment to specify which model will gain better results compared to others; a set of enhancements can be advised and critical issues can be denoted to obtain better results. Accordingly, different performance measures were used to judge proposed method's performance:

1- The Accuracy (AC)

It is the amount of the total correct predictions to the actual data set size. It can be determined by applying equation (1): (Elhamahmy et al., 2010)

$$AC = \frac{TP+TN}{TP+TN+FP+FN} \qquad \dots (5.1)$$

Where:

- True Positive (TP) is the amount of attack detected when it is actually attacked. (Gaidhane, et.al, 2014).

- True Negative (TN) is the amount of normal detected when it is actually normal.

- False Positive (FP) is the amount of attack detected when it is actually normal called in which it called a false alarm.

- False Negative (FN) is the amount of normal detected when it is actually attacked, namely the attacks which can be detected by an intrusion detection system.

2- Detection Rate (DR)

It can be defined as "the ratio between the number of correctly detected attacks and the total number of attacks"(Kumar, 2014).

As shown in equation (2):

$$DR = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \qquad \ldots(5.2)$$

3- False Alarm Rate (FAR)

It can be defined as" the number of 'normal' patterns classified as attacks (False Positive) divided by the total number of 'normal' patterns"(Elhamahmy, et al., 2010). As shown in equation (3):

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} \qquad \ldots(5.3)$$

To provide good judgment on the proposed work we compared his method results with different results from a set of previous studies that used an XCS to IDS.

These studies are:

1. The role of early stopping and population size in XCS for intrusion detection (Shafi, K., Abbass, et.al, 2006).
2. A fuzzy-genetic approach to network intrusion detection (Fries, T. P. ,2008).
3. Applying Artificial Neural Network and eXtended Classifier System for Network Intrusion Detection (Alsharafat Wafa, 2010).
4. Computer intrusion detection by two-objective fuzzy genetic algorithm (Agravat, M., and Rao, U. P. 2011).
5. Intelligent Detection of Intrusion into Databases Using Extended Classifier System (Yazdani, 2013).
6. Improved Detection of Intrusion to Computer Networks using Extended Classification Systems (Panahi, 2013).
7. A Novel Hybrid Model for Network Intrusion Detection (Tiwari, 2013).
8. An effective rule generation for Intrusion Detection System using Genetics Algorithm (Kadam, 2013).
9. Selecting GA Parameters for Intrusion Detection (Pawar, 2014).
10. Predictive Rule Discovery for Network Intrusion Detection (Patel, 2015).
11. Intrusion Detection System using Fuzzy Genetic Algorithm (Danane, 2015).
12. Intrusion detection using feature selection and machine learning algorithm with misuse detection (Sasan and Sharma, 2016).

Data Collection

To perform experimental results the KDD’99 dataset was used. The KDD’99 dataset comprises a set of 41 features; each feature is a result of a connection and a label specifying the status of connection records whether normal or specific attack type. KDD dataset includes training and testing record sets. The total number of connection records in the training dataset is about 5 million records (SJ, et al. 2011).

KDD 99 is the most suitable dataset benchmark of reference to be used in experiments. Various researchers have used KDD'99 to validate their results. In the scope of this study, 10% of KDD'99 will be used to train and test, which the 10% of KDD'99 represents a normal distribution of KDD'99 that consists of 500,000 network packets, each called a record. The records in the KDD '99 dataset, contain information about 41 network packets (Farid, et al., 2009) (Table 5.1).

Table 5.1: Features of the KDD '99 Dataset

Feature Number	Feature Name	Type (1)	Description
1	Duration	C	length (number of seconds) of the connection
2	Protocol type	D	type of the protocol, e.g. tcp, udp, etc.
3	Service	D	network service on the destination, e.g., http, telnet, etc.
4	Flag	D	normal or error status of the connection
5	Src_bytes	C	number of data bytes from source to destination

6	Dst_bytes	C	number of data bytes from destination to source
7	Land	D	1 if connection is from/to the same host/port; 0 otherwise
8	Wrong fragment	C	number of ``wrong'' fragments
9	Urgent	C	number of urgent packets
10	Hot	C	number of ``hot'' indicators
11	Num_failed_logins	C	number of failed login attempts
12	Logged in	D	1 if successfully logged in; 0 otherwise
13	Num_compromised	C	number of ``compromised'' conditions
14	Root shell	C	1 if root shell is obtained; 0 otherwise
15	Su_attempted	C	1 if ``su root'' command attempted; 0 otherwise
16	Num_root	C	number of ``root'' accesses
17	Num_file_creations	C	number of file creation operations
18	Num_shell	C	number of shell prompts
19	Num_access_files	C	number of operations on access control files
20	Num_outbound_cmds	C	number of outbound commands in an ftp session
21	Is_host_login	D	1 if the login belongs to the ``hot'' list; 0 otherwise
22	Is_guest_login	D	1 if the login is a ``guest'' login; 0 otherwise
23	Count	C	number of connections to the same host as the current connection in the past two seconds
24	Srv_count	C	number of connections to the same service
25	Serror_rate	C	% of connections that have ``SYN'' errors
26	Srv_serror_rate	C	% of connections that have ``SYN'' errors

27	Rerror_rate	C	% of connections that have ``REJ" errors
28	Srv_rerror_rate	C	% of connections that have ``REJ" errors
29	Same_srv_rate	C	% of connections to the same service
30	Diff_srv_rate	C	% of connections to different services
31	Srv_diff_host_rate	C	% of connections to different hosts
32	Dst_host_count	C	count for destination host
33	Dst_host_srv_count	C	srv_count for destination host
34	Dst_host_same_srv_rate	C	same_srv_rate for destination host
35	Dst_host_diff_srv_rate	C	diff_srv_rate for destination host
36	Dst_host_same_src_port_rate	C	same_src_port_rate for destination host
Feature Number	Feature Name	Type (1)	Description
37	Dst_host_srv_diff_host_rate	C	diff_host_rate for destination host
38	Dst_host_serror_rate	C	serror_rate for destination host
39	Dst_host_srv_serror_rate	C	srv_serror_rate for destination host
40	Dst_host_rerror_rate	C	rerror_rate for destination host
41	Dst_host_srv_rerror_rate	C	srv_serror_rate for destination host
42	Attack name	-	-
(1) C: Continuous; D: Discrete.			

Experiments

Here, we divided the work into two experiments; at first; 41 features were used in the proposed system model and then results were recorded. Furthermore; a FLAME features filtration algorithm was implemented to reduce the number of features from 41 to 20.

Experiment 1

Parameters Setting

In this study, a set of parameters must be determined before conducting experiments in terms of finding optimal or near optimal solutions. These parameters include:

1. Crossover Probability.
2. Mutation Probability.
3. A number of optimizations (Generation).
4. Number of solutions (Nests).
5. Abandoned Probability.

Crossover Probability

We used uniform crossover with different crossover probability. The experiment results are listed in Table 5.2. *

Note: * see Appendix A, Table5.

Table 5.2: Crossover Probability and Number of optimizations (Generation)

Crossover probability	DR%	ACC %	Number of optimizations (Generation)	Number of solutions (Nests)	Abandoned Probability
0.1	76.9695	69.2725	10	50	0.3
0.1	84.7100	76.2390	200	50	0.3
0.2	89.1802	80.2622	400	50	0.3
0.7	93.7005	84.3305	1000	50	0.3
0.9	82.1598	73.9438	100	50	0.3

The table **(**5.2) shows that high detection rate was arrived to 93.70, with an accuracy of 84.33. We obtained a high detection rate when the value of crossover is 0.7 and 1000 generations. The results also recorded when used 50 solutions (Nest) and the value of abandoned probability equal 0.3.

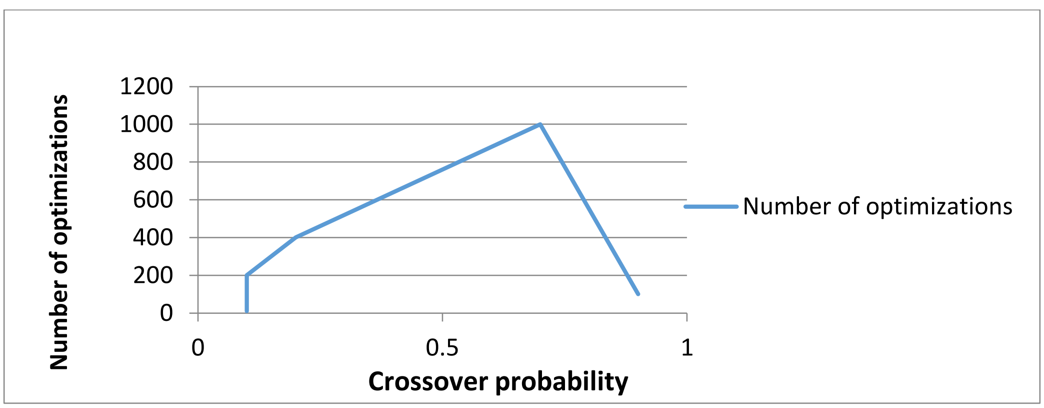


Figure (5.1): Crossover Probability and Number of optimizations (Generation)

The figure (5.1) shows that we acquired a DR when the value of crossover is 0.7 and 1000 generations.

Note: * see Appendix A, Table5.

5.4.1.1.2 Mutation Probability

We used a random mutation along with different mutation probabilities. The experiment showed that when number of generations increases, the DR and ACC are growing and becoming higher than others as listed in Table 5.3. *

Table 5.3: Mutation Probability and Number of optimizations (Generation)

Mutation probability	DR%	ACC%	Number of optimizations (Generation)	Number of solutions (Nests)	Abandoned Probability
0.1	82.1598	73.9438	100	50	0.3
0.1	84.7100	76.2390	200	50	0.3
0.1	93.7005	84.3305	1000	50	0.3
0.2	76.9695	69.2725	10	50	0.3
0.2	89.1802	80.2622	400	50	0.3

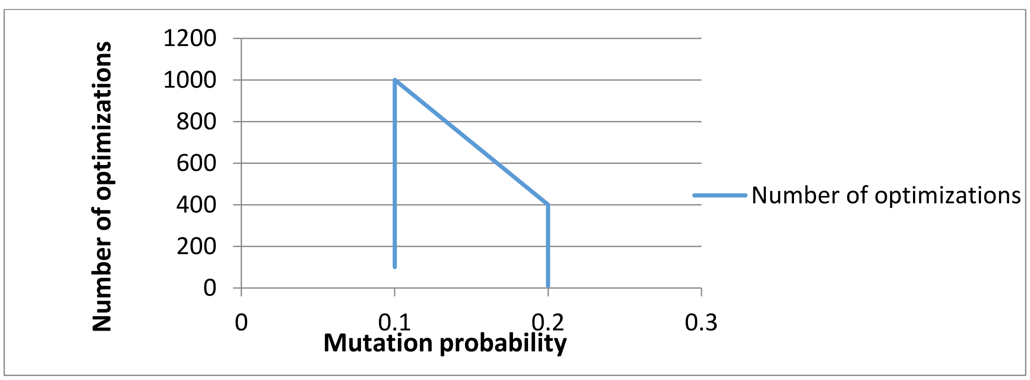


Figure (5.2): Mutation Probability and Number of optimizations (Generation)

The table(5.3) and Figure (5.2) shows that the highest DR was recorded is 93.70% with an ACC of 84.33% considering a p*m* within the value of 0.1 and 1000 generations accompanied with 50 solutions and an abandoned probability of 0.3.

Note: * see Appendix A, Table5.

Number of optimizations (Generation)

It is important to find the number of optimizations (Generations) for optimal solutions for an XCS for network intrusion detection.

In the thesis, the experiment showed that when the number of generations increases, the DR is growing and becomes higher than others. As shown in table 5.4. *

Table 5.4:Detection Rate with Number of optimizations (Generation)

Number of optimizations(Generation)	DR%
10	77.3235
20	79.4896
30	80.1989
50	81.4203
100	82.7026
150	84.5557
200	86.0793
250	86.4053
300	87.0433
400	88.9923
500	89.1066
1000	93.2519

The table (5.4) shows that high performance for DR is detected with high ACC when 1000 generations are imbedded with abandoned Probability=0. 3.

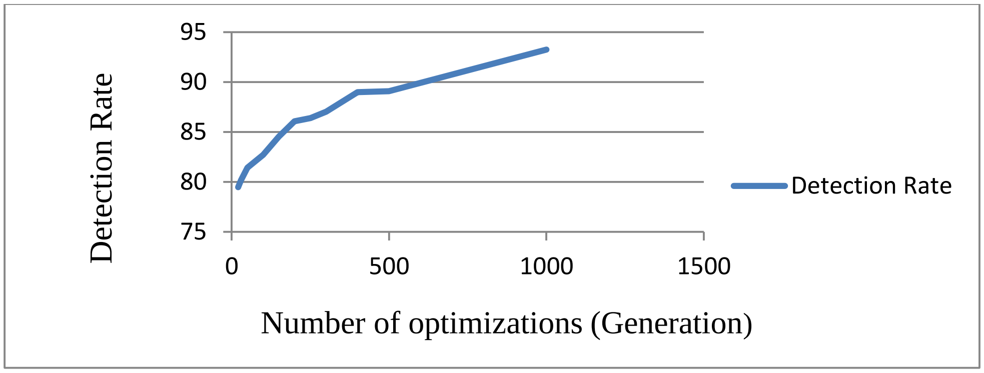


Figure (5.3): Detection Rate with Number of optimizations (Generation)

The results show that when increased the number of optimizations (Generation) the detection rate was increased and obtained high DR at 1000 generations as shown in figure (5.3).

Note: * see Appendix A, Table3.

Table 5.5: Accuracy with Number of optimizations (Generation)

Number of optimizations Generation)	**ACC%**
10	69.5911
20	71.5406
30	72.1790
50	73.2783
100	74.4323
150	76.1001
200	77.4713

250	77.7648
300	78.3390
400	80.0931
500	80.1959
1000	83.9267

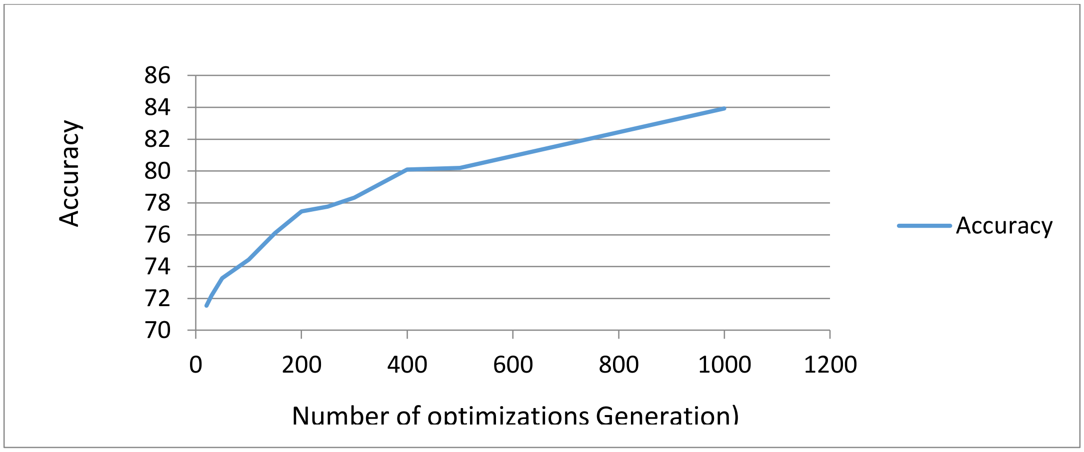


Figure (5.4):.Accuracy with Number of optimizations (Generation)

When increased the number of optimizations (Generation) the ACC was increased and obtained high ACC around 83.92% with 1000 generations as shows in table(5.5) and figure (5.4).

Number of solutions (Nests)

The applied experiment showed that when the number of solutions increases the detection rate is growing and becomes higher than others. We can show results in Table (5.6). *

Note: * see Appendix A, Table5.

Table 5.6: Detection Rate with Number of solutions (Nests)

Number of solutions (Nests)	**DR%**
10	93.2519
50	93.0848
150	93.3457

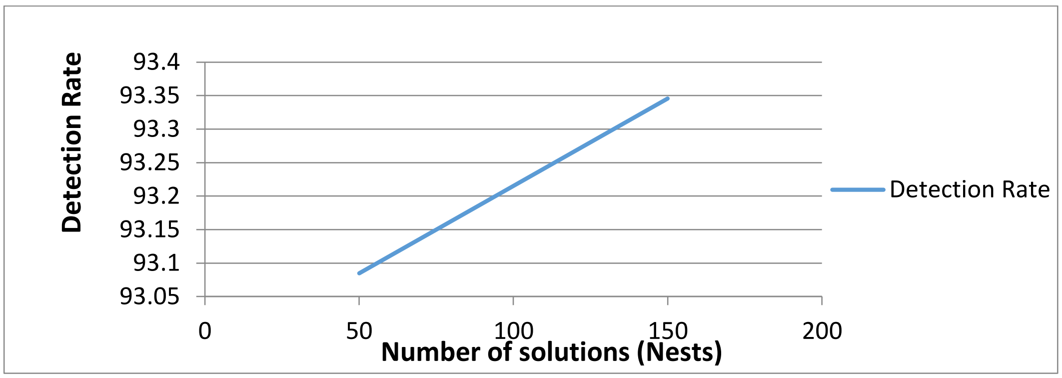


Figure (5.5): Detection Rate with Number of solutions (Nests)

The table (5.6) and figure(5.5) shows that when increased the number of solutions(Nests) the detection rate not affect, the values of detection rate around 93% and obtained a high detection rate at 1000 generations.

The experiment proves that when the number of solutions increases, the accuracy is growing and becomes higher than others (Table 5.7). *

Table 5.7:Accuracy with Number of solutions (Nests)

Number of solutions (Nests)	ACC%
10	83.9267
50	83.7763
150	84.0111

Note: * see Appendix A, Table (3,4,5).

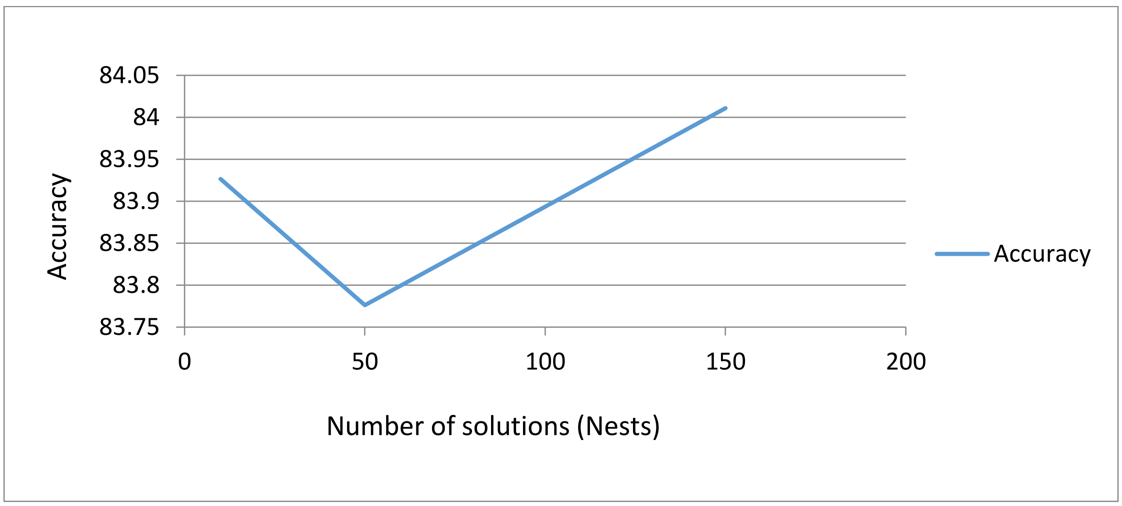


Figure (5.6): Accuracy with Number of solutions (Nests)

The results show that when increased the number of solutions(Nests) the ACC not affect, the values of ACC around 83% and obtained an accuracy rate at 150 solutions as shows in the table (5.7) and figure(5.6).

Abandoned Probability

In the scope of this study, the experiment showed that when the value of the abandoned probability was increase the DR is growing and becomes higher than others (Table 5.8).

Table 5.8: Detection Rate with Abandoned Probability

Abandoned Probability	DR%
0.1	93.2823
0.2	93.0848
0.3	93.7005

The table (5.8) shows that high performance for detection rate is recorded when abandoned Probability equals 0.3.

The experiment showed that when abandoned probability increases, the accuracy is growing and becomes higher than others (Table 5.9).

Table 5.9: Accuracy with Abandoned Probability

Abandoned Probability	ACC%
0.1	84.0111
0.2	83.7763
0.3	84.3305

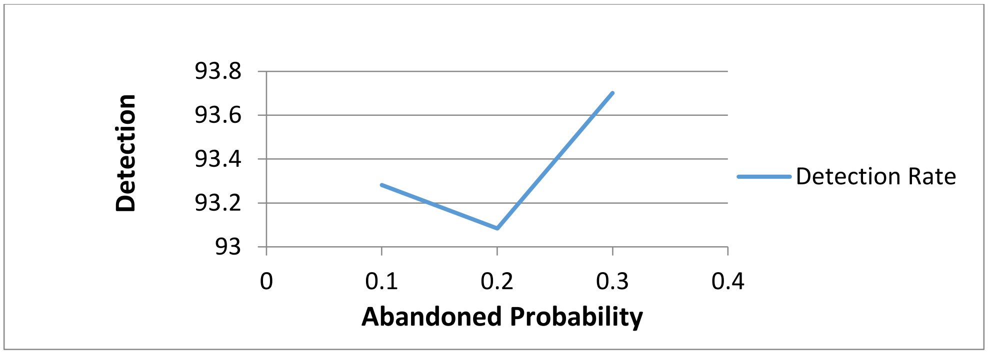


Figure (5.7): Detection Rate with Abandoned Probability

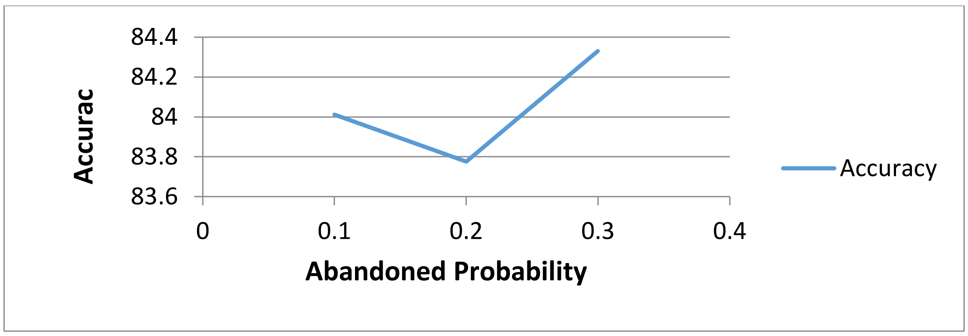


Figure (5.8): Accuracy with Abandoned Probability

The Table (5.9) and figure (5.7), (5.8) show that the values of the detection rate and accuracy increased small values when abandoned probability increased.

Experiment 2:FLAME Features Flirtation

We used FLAME within this study; FLAME is a clustering algorithm that defines clusters in the slow parts of a dataset and performs the cluster assignment solely based on the neighborhood relationships among objects.

The main characteristic of this algorithm is that the neighborhood relationships among neighboring objects in the feature space are applied to tighten up the memberships of neighboring objects in the fuzzy membership space.

In the proposed model; a set of parameters must be determined while conducting experiments to assure the finding of optimal or near optimal solutions; these parameters are:

1. Crossover Probability.
2. Mutation Probability.
3. Number of optimizations (Generation).
4. Number of solutions (Nests).
5. Abandoned Probability.
6. Number of features.

Crossover Probability

We used uniform crossover coupled with different values of crossover probabilities. The experiment (filtration features) showed that when the crossover probability increases, the detection rate and accuracy are growing and become higher than others as noted in Table (5.10). *

Note: * see Appendix B, Table 9.

Table 5.10: Crossover probability, Detection Rate, Accuracy and False alarm rate

Crossover probability	DR%	ACC%	FAR%
0.6	SY: 99.98815	99.9180	0.0118
0.7	SY: 99.9839	99.9015	0.0160
٠.٩	SY: 99.9950	99.8938	0.0049

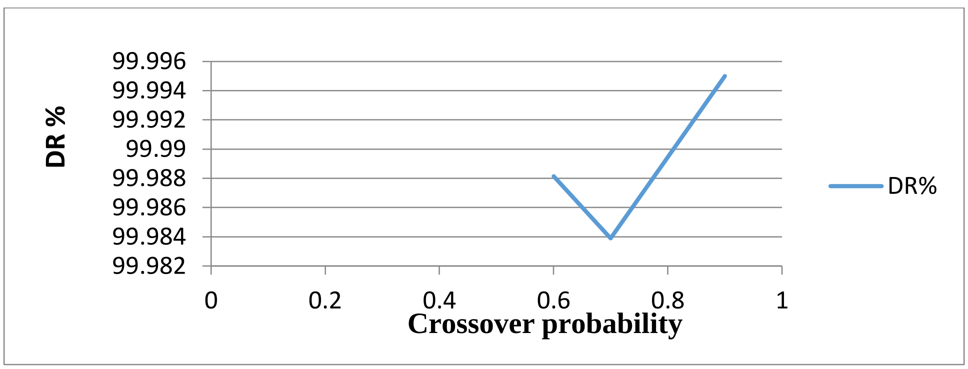


Figure (5.9): Crossover probability, Detection Rate

The table (5.10) and figure (5.9) shows that high performance for DR equal to 99.99 and accuracy equal to 99.89% when the crossover probability equal 0.9 .

Mutation Probability

The researcher used random mutation with different mutation probabilities. The experiment (filtration features) showed that when a mutation probability is detected by a low value, the detection rate and accuracy are growing and become higher orders (Table 5.11). *

Note: * see and Appendix B, Table 9.

Table 5.11: Mutation probability, Detection Rate, Accuracy and False alarm

Mutation probability	DR%	ACC%	FAR%
0.1	SY: 99.9881	99.9180	0.0118
0.2	SY: 99.9839	99.9015	0.0160
٠.٣	SY: 99.99507	99.8938	0.0049

The table (5.11) shows that high performance of for DR equal 99.99% accompanied with an accuracy of 99.89% and FAR=0.0049% is detected when the mutation probability is equal to 0.3.

A Number of optimizations (Generation) / Number of Features

When using the FLAME algorithm the experiment showed that a number of optimizations decrease compared to the operation before using FLAME and the detection rate and accuracy are growing and become higher than others.

Also, the attacks were divided into four classes; DoS, R2L, U2R and Probe. The table 5.12 shows the attack types and sizes.

Table 5.12: Attack types and the sizes.

Attack Type	Size of Data
DOS	51%
Probe	28%
R2L	11%
U2R	8%
Normal	2%

Also, each attack class possesses a specific action as listed in Table 5.13.

Table 5.13: List of attacks (category wise)

Attack class	DoS	R2L	U2R	Probe
Attack Name	Back, land ,Neptune, pod smurf ,teardrop	ftp_write ,guess_passwd ,imap ,multihop, phf spy, warezclient ,Warezmaster	buffer_overflow , loadmodule, perl rootkit	ipsweep ,nmap portsweep ,satan

Each attack has its own features as follows. DoS attack specific features numbers and names are listed Table 5.14.

Table 5.14: Features Selected for DoS, Probe, U2R, R2L attacks

Features Selected for	**Feature Number**
DoS	1,2,14,9,10,11,8,20
Probe	12,7,3,4,23,24,21
U2R	13,5,25
R2L	6,22

The best detected values for the number of optimizations was correlated with high detection rates as listed in Tables (5.15, 5.16 and 5.17). *

Table 5.15:Number of optimizations, Detection Rate, Accuracy and False alarm rate for selected 18 features

Number of optimizations	**DR%**	**ACC%**	**FAR%**	**Number of solutions (Nests)**	**Number of Selected Features**	**Index Features**
10	DOS:99.9708	99.9708	0.0291	50	18	19,13 ,16, 41, 3 4, 2, 15, 9, 14, 17, 18, 1, 12, 11 33, 32, 27
	Probe:99.5604	87.0230	0.4395			
	R2L:72.2394	72.2394	27.7605			
	U2R:99.9982	99.9982	0.0017			
	SY: 92.9422	89.8078	7.0577			
100	DOS:99.9511	99.9511	0.0488	50	18	13, 16, 41, 15, 3 4, 2, 9, 14, 17, 19, 20, 18, 1, 12 11, 33, 32
	Probe: 98.0446	2.1895	1.9553			
	R2L: 42.7044	32.5761	57.2955			
	U2R: 99.9873	99.9873	0.01269			
	SY: 85.1718	58.6760	14.8281			

Table 5.15 shows that high performance value of 92.94 for detection rate correlates with an accuracy of 0.80% and false alarm rate of 7.06% considering a number of optimizations of 10.

Note: * see Appendix B, Table (6,7,8).

Table 5. 16: Number of optimizations, Detection Rate, Accuracy and False alarm rate for

selected 18 features

Number of optimizations	DR%	ACC%	FAR%	Number of solutions (nests)	Number of selected features	Index Features
10	DOS:49.9916	2.6998	50.0083	150	18	13,16,20,19 15, 3, 4,2,9,41,14 17,18,1,12,33, 11,32
	Probe:99.9999	99.9999	0.003			
	R2L: 51.8403	48.4227	48.1596			
	U2R: 99.9887	99.9887	0.01126			
	SY: 75.4551	62.7778	24.5448			
100	DOS: 99.9940	99.9940	0.0059	150	18	13,16,2,3,4,19 20,15,9,14,17 41,18,1,12,32 33,11
	Probe:99.94171	99.6123	0.0582			
	R2L: 100.0	100.0	0.0			
	U2R: 99.9999	99.9999	0.0051			
	SY: 99.9839	99.9015	0.0160			
1000	DOS: 98.2817	98.2817	1.7182	150	18	3, 4, 2, 13, 16 15, 20, 19, 41, 9,14, 17,18, 1 12, 11, 32 ,33
	Probe:99.9428	99.0326	0.0571			
	R2L: 99.9999	99.9999	0.0055			
	U2R: 99.9999	99.9999	0.0015			
	SY: 99.5561	99.3285	0.4438			

The table (5.16) shows that high performance of 99.98% for detection rate happens with an accuracy of 99.90% and false alarm rate equal of 0.016% when the number of optimizations is 100.

Table 5.17: Number of optimizations, Detection Rate, Accuracy and False alarm rate for

Selected 20 features

Number of optimizations	DR%	ACC%	FAR%	Number of solutions (nests)	Number of selected features	Index Features
10	DOS: 99.9810	99.9810	0.0189	50	20	1, 2, 12, 13, 14, 9, 10 11, 6, 7, 8, 3, 4, 5, 23 24, 25, 20, 21, 22
	Probe:99.9878	98.6380	0.0121			
	R2L: 100.0	100.0	٠.٠			
	U2R: 99.9999	99.9999	0.0291			
	SY: 99.9922	99.6547	0.0077			
100	DOS: 99.9999	99.9999	0.0065	50	20	4, 5, 1, 2 ,3 ,14, 15, 16, 11, 12, 13, 9, 10, 6, 7, 8, 24, 25, 26, 22
	Probe:99.9813	99.7764	0.0186			
	R2L: 100.0	100.0	٠.٠			
	U2R: 99.9989	99.7990	0.00102			
	SY: 99.9950	99.8938	0.00492			

Table 5.17 shows that high performance of 99.99% for DR corresponds to an ACC of 99.89% and FAR of 0.005% when the number of optimizations is 100.

Table 5.18: Number of optimizations, Detection Rate Accuracy, False alarm rate for

selected 20 features

Number of optimizations	DR%	ACC%	FAR%	Number of solutions (nests)	Number of features	Index Features
10	DOS:99.9781	99.9781	0.02186	150	20	7, 8, 9, 4, 5, 6,2, 3, 1 19, 20,16,17,18,13 14,12,10, 11, 15
	Probe:99.9999	99.9999	0.0061			
	R2L:99.9999	99.9999	0.0003			
	U2R:99.9999	99.9999	0.0015			
	SY: 99.99453	99.9945	0.0054			
100	DOS:99.3223	92.2794	0.6776	150	20	32, 33, 34, 29, 30, 31, 27, 28, 24, 25, 26, 41, 40, 38, 39, 35, 36, 37, 1, 2
	Probe:96.6375	95.1256	3.3624			
	R2L: 99.9999	99.9999	0.0019			
	U2R:99.9968	99.9968	0.0031			

	SY: 98.9891	96.8504	1.0108			
1000	DOS: 99.5469	70.9175	0.45308	150	20	19, 20, 21, 17, 18, 14, 15, 16, 11, 12, 13, 29, 30, 31, 27, 28, 24, 25, 26, 22
	Probe:99.9945	99.6806	0.0054			
	R2L: 100.0	100.0	٠.٠			
	U2R:99.9695	99.8270	0.0304			
	SY: 99.8777	92.6063	0.1222			

Table 5.18 shows that 99.98% high performance value for DR goes along with an ACC of 99.90% and FAR of 0.005% when the number of optimizations equals 100.

Table 5.19: Number of optimizations, Detection Rate, Accuracy and False alarm for selected 25 features

Number of optimizations	DR%	ACC%	FAR%	Number Of solutions (Nests)	Number Features	Index Features
10	DOS: 99.3206	89.9129	0.6793	50	25	38, 25, 37, 5,3, 4, 2 28, 20, 33, 32,35, 23 24, 21,7,12,27,26,40 30,39, 29, 31, 36
	Probe:99.9999	99.9999	0.0047			
	R2L: 99.9999	99.9999	0.0070			
	U2R: 100.0	100.0	0.0			
	SY: 99.8301	97.4782	0.1698			
100	DOS: 99.9566	99.7570	0.0433	50	25	38, 25, 37, 5, 41, 28 33, 32,4,2,3,35,23 24,7,21,12,20,27,26 40,30,39,29,31
	Probe:99.9982	99.9175	0.0017			
	R2L: 100.0	100.0	0.0			
	U2R:99.9977	99.9977	0.0022			
	SY: 99.9881	99.9180	0.0118			

Table (5.19) shows that a high performance of 99.98% for DR is accompanied with an ACC of 99.91% and 0.011% FAR when the number of optimizations is 100.

Table 5.20:Number of optimizations, Detection Rate, Accuracy, False alarm for

selected 25 features

Number of optimizations	DR%	ACC%	FAR%	Number of solutions (nests)	Number of selected features	Index features
10	DOS: 99.6214	93.3247	0.3785	150	25	38, 2, 25,37, 41,5,3,4,20, 28,33,21,32, 35,23,24,7,12, 27,26,40,30, 39,29,31
	Probe:99.8469	99.8469	0.1530			
	R2L: 100.0	100.0	0.0			
	U2R: 99.999	99.9996	0.00034			
	SY: 99.8670	98.2928	0.1329			
100	DOS: 99.9688	99.6859	0.0311	150	25	41,38,25,37,5, 4,2,3,28,21, 20,33,32,35, 23,24,7,12,27, 26,40,30,39 29,31
	Probe: 99.8651	99.8318	0.1348			
	R2L: 99.99999	99.6770	0.0096			
	U2R: 99.9954	99.9954	0.0045			
	SY: 99.95737	99.7975	0.0426			
1000	DOS: 99.9958	99.9958	0.0041	150	25	38,25,37,5,41, 28,2,3,33,32, 4,20,35,21,23, 24,7,12,27,26, 40,31,39,29, 30
	Probe:99.8081	87.57000	0.1918			
	R2L: 99.9999	99.99993	0.0060			
	U2R: 99.9999	99.99998	0.0010			
	SY: 99.95097	96.89143	0.04902			

Table (5.20) shows that high performance for DR equals 99.95% when the ACC equals 99.79% with FAR of 0.042% and a number of optimizations of 100.

Number of solutions (Nests)

The experiment applied in this study has proven that after using FLAME; a number of solutions the same value compared to solutions achieved before the implementation of FLAME in which the DR and ACC are growing and become higher than others. The best value obtained was for solutions number of 50 with variable feature numbers as listed in tables (5.21,5.22,5.23).*

Note: * see Appendix B, Table (6,7,8).

Table 5.21:Number of solution optimizations, Detection Rate, Accuracy and False alarm for selected 18 features

DR%	ACC%	FAR%	Number of solutions (Nests)	Number of selected features	Index features
DOS:99.9708	99.9708	0.0291	50	18	19,13,16,41 3,4,2,15,9 14,17,18,1 12,11,33,32 27
Probe:99.5604	87.0230	0.4395			
R2L:72.2394	72.2394	27.7605			
U2R:99.9982	99.9982	0.00176			
SY: 92.9422	89.8078	7.05775			
DOS: 99.99403	99.9940	0.00599	150	18	13,16,2,3,4 19,20,15,9 14,17,41, 18,1,12,32 33,11
Probe:99.9417	99.6123	0.0582			
R2L: 100.0	100.0	0.0			
U2R: 99.9999	99.9999	0.0051			
SY: 99.98394	99.9015	0.0160			

Table 5.21 shows that high performance for DR of 99.98% is found with an ACC of 99.91% and FAR of 0.016% when the number of solutions is 100 considering 18 features.

Table 5.22:Number of solution optimizations, Detection Rate, Accuracy and False alarm rate for selected 20 features

DR%	ACC%	FAR%	Number of solutions (nests)	Number of selected features	Index features
DOS: 99.9999	99.9999	0.0065	50	20	4,5,1, 2,3, 14, 15 16, 11, 12,13 ,9 10, 6 ,7 ,8 ,24,25 26, 22
Probe:99.9813	99.7764	0.0186			
R2L: 100.0	100.0	٠.٠			
U2R: 99.9989	99.7990	0.00102			
SY: 99.9950	99.8938	0.00492			

DOS:99.9781	99.9781	0.02186	150	20	7, 8 ,9, 4, 5, 6, 2 3, 1, 19, 20, 16 17, 18, 13, 14, 12, 10, 11, 15
Probe:99.9999	99.9999	0.0061			
R2L:99.99999	99.9999	0.0003			
U2R:99.99999	99.99999	0.0156			
SY: 99.994533	99.9945	0.00546			

Table 5.22 shows that high performance for DR is 99.99% when the ACC equals 99.89% with FAR of 0.005% and a number of solutions of 50 with 20 features.

Table 5.23:Number of solution optimizations, Detection Rate, Accuracy and False alarm for selected 25 features.

DR%	ACC%	FAR%	Number of solutions (Nests)	Number Features	Index Features
DOS: 99.9566	99.7570	0.0433	50	25	38, 25, 37, 5, 41, 28, 33, 32, 4,2 3, 35, 23, 24, 7 ,21, 12 20, 27 26, 40, 30 ,39, 29, 31
Probe:99.9982	99.9175	0.00171			
R2L: 100.0	100.0	0.0			
U2R:99.9977	99.9977	0.00228			
SY: 99.9881	99.9180	0.01184			
DOS: 99.9688	99.6859	0.0311	150	25	41, 38, 25, 37, 5, 4, 2, 3, 28, 21 20, 33, 32 ,35, 23, 24, 7, 12, 27 26, 40, 30, 39, 29, 31
Probe: 99.8651	99.8318	0.1348			
R2L: 99.99999	99.6770	0.0096			
U2R: 99.9954	99.9954	0.00455			
SY: 99.9573	99.7975	0.04262			

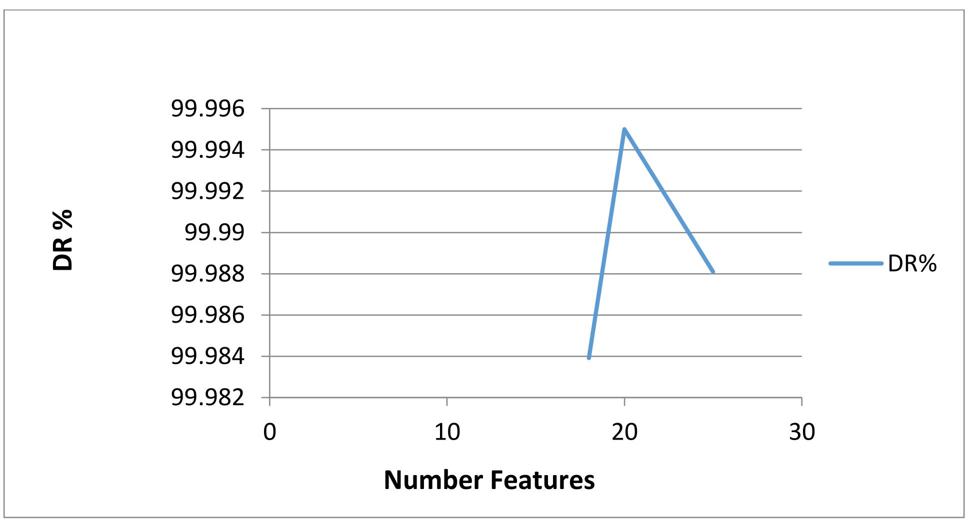
Table 5.23 shows that high performance for DR equal to 99.98%,ACC equal to 99.91% and FAR equal to 0.011% when a number of solutions equal to 50 with 25 features.

roposed Solution System

In this thesis, the results prove that the proposed system model obtains the highest detection rate of 99.99 when the numbers of features equals 20 (Table 5.24).

Table 5.24:Number features, Detection Rate, Index Features

Number Features	DR%	Index Features
18	99.9839	13, 16, 2 ,3 ,4 ,19,20, 15, 9, 14 ,17, 41, 18, 1, 12, 32, 33 ,11
20	99.9950	4, 5, 1, 2, 3 ,14 ,15, 16, 11, 12, 13, 9, 10, 6, 7, 8, 24, 25, 26, 22
25	99.9881	38, 25, 37, 5, 41, 28, 33, 32, 4, 2, 3, 35, 23, 24, 7, 21, 12, 20, 27, 26, 40, 30, 39, 29, 31



Figure(5.10): Number features, Detection Rate

The results in table (5.24) and figure (5.10) prove that the proposed system model obtains the highest DR of 99.99% when the numbers of features equals 20.

The results also showed that the proposed system model achieves the highest ACC of 99.89% when the numbers of features is 20, as shows in Table (5.25).

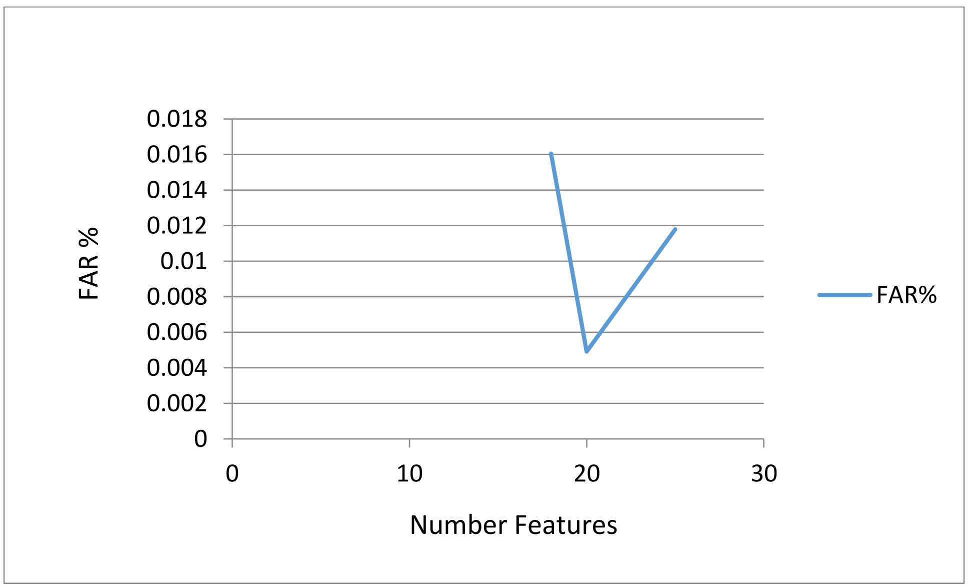
Table 5.25: Number Features and Accuracy

Number Features	ACC%	Index Features
18	99.9015	13, 16, 2, 3, 4, 19, 20 , 15, 9, 14, 17 ,41, 18, 1, 12 ,32 ,33 11
20	99.8938	4, 5 ,1, 2, 3, 14 ,15, 16 ,11, 12 ,13, 9, 10, 6, 7 ,8 ,24 ,25, 26, 22
25	99.9180	38 ,25, 37, 5, 41, 28, 33, 32, 4, 2, 3, 35, 23, 24, 7, 21, 12, 20, 27, 26, 40, 30, 39, 29, 31

Furthermore, the results show that the proposed system model reaches its highest FAR value of 99.89% when the number of features is 20 as proven in Table (5.26).

Table 5.26: Number Features, False alarm and Index Features.

Number Features	**FAR%**	**Index Features**
18	0.01605	13, 16, 2 ,3, 4 19 ,20, 15 ,9 ,14 ,17, 41, 18, 1, 12, 32, 33, 11
20	0.00492	4, 5, 1, 2, 3, 14, 15, 16, 11, 12, 13, 9, 10 ,6, 7, 8, 24, 25 ,26 ,22
25	0.0118	38, 25, 37, 5, 41, 28, 33, 32, 4, 2, 3, 35, 23, 24, 7, 21, 12, 20, 27, 26, 40, 30, 39, 29, 31



Figure(5.11): Number Features, False alarm rate

The table (5.26) and figure (5.11) shows that the small value of FAR when used 20 features.

Finally, the results show that the proposed system model obtains its highest DR and ACC along with low FAR when the number of optimizations equals 100 with a number of solutions of 50 as shows in Table 5.27.

Table 5.27: Number of optimizations, Accuracy and False alarm rate

Number of optimizations	**DR%**	**ACC%**	**FAR%**	**Number of solutions (nests)**	**Number of selected features**	**Index features**
100	DOS: 99.9999	99.9999	0.0065	50	20	4, 5, 1 ,2,3, 14 ,15, 16, 11, 12, 13, 9, 10 6 ,7 ,8 ,24, 25 ,26 ,22
	Probe:99.9813	99.7764	0.0186			
	R2L: 100.0	100.0	٠.٠			
	U2R: 99.9989	99.7990	0.00102			
	SY: 99.9950	99.8938	0.00492			

Impact Before and After Using the filtration (FLAME) process

When comparing the results before the of filtration when we used 41 features and After filtration; the features were reduced from 41 to 20. The effect posed by the implementation of FLAME algorithm is listed in Table (5.28).

Table 5.28: Parameters, Before filtration and After filtration (FLAME)

Parameters	Before filtration	After filtration (FLAME)
Crossover Probability	0.7	0.9
Mutation Probability	0.1	0.3
Number of optimizations (Generation)	1000	100
Number of solutions(Nests)	50	50
Number Features	41	20
Detection Rate	93.7005%	99.9950%
Accuracy	84.3305%	99.8938%
False Alarm Rate	6.2994%	0.0049%

Table 5.28 shows that the crossover probability increased before the implementation of FLAME from 0.7 to 0.9 and the mutation probability increased from 0.1 to 0.3. FLAME implementation decreased the number of generations from 1000 to 100 which is better for the system with a DR range of 93.70% to 99.99% along with an ACC range from 84.33% to 99.89% and a low FAR range from 6.29% to 0.004%.

Comparative Study

A comparative study of several studies and this thesis' proposed method is presented in Table 5.29.

Table 5.29: comparative study of the methods, the number features and detection rate

Method	**Number Features**	**DR%**
Proposed work	20	99.99
Alsharafat, 2010 Artificial Neural Network (ANN) extended Classifier System	-	98.01
Panahi, 2013 Data mining and extended classification system	41	94.83
Yazdani, et.al, 2013 eXtended classifier systems (XCS)	-	91
Fries, Terrence P (2008) A fuzzy-genetic approach to network intrusion detection	8	99.6

Table 5.29 indicates that the DR of this proposed work got higher values compared to other studies .

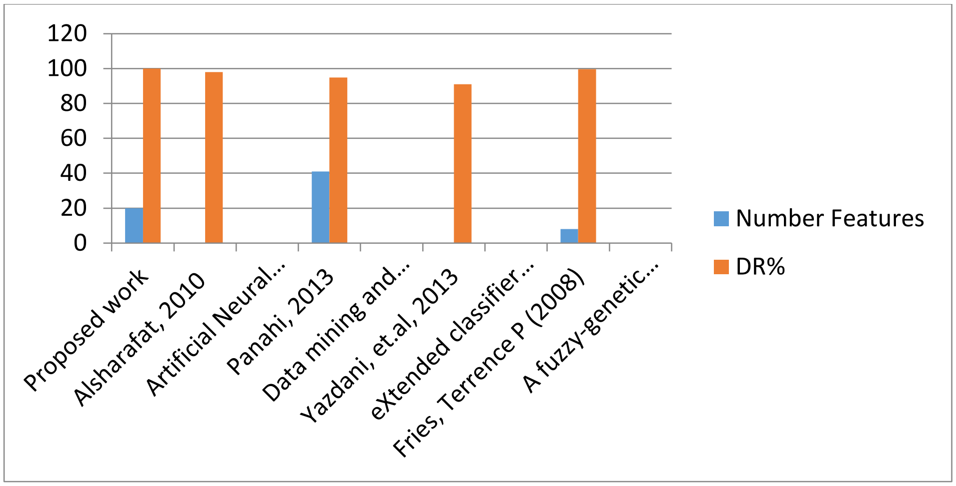


Figure (5.12): comparative study of the methods, the number features, detection rate

The figure (5.12) indicates that the DR of this proposed work got higher values compared to other studies.

Table 5.30: comparative study of the methods, the number features, accuracy

Method	Number Features	ACC%
Proposed work	20	99.89
Shafi, A. Abbass, 2006 The Role of Early Stopping and Population Size in XCS for Intrusion Detection.)	29	95
Agravat, Rao, (2011) fuzzy Genetic-based Learning algorithms	20	98.5

Table 5.30 indicates that the accuracy of this proposed work got higher values compared to other studies.

Shrivastava, Jain (2011) Rough Set Theory and support vector machine	6	95.98
Sasan and Sharma (2016) Intrusion detection using feature selection and machine learning algorithm with misuse detection	29	88.23

Table 5.31: Methods, Features Number, Detection Rate values for Each Type of Attacks

Method	**Number Features**	**Detection Rate DOS %**	**Detection Rate Probe %**	**Detection Rate R2L %**	**Detection Rate U2R %**
Proposed work	20	99.99	99.98	100	99.99
Alsharafat, 2010 Artificial Neural Network (ANN) extended Classifier System (XCS)	-	98.8	90.5	34.6	88.5
Te-Shun C., 2007, Ensemble Fuzzy Belief Intrusion Detection Design	-	99.86	95.52	0	0
Tiwari, 2013 Firefly algorithm (FA) feature selection, Radial basis function (RBF), Rough Set Theory	32	99.0	98.0	97.0	95.0

The table 5.31 shows that this proposed work achieves higher detection rates for each type of attacks. Tiwari (2013) study contains the nearest values of this proposed work. The better performance of Tiwari study was a result of the application of firefly algorithm and neural network, but, Tiwari used 32 features while this proposed work used 20features.

Table 5.32: Methods, Number Features and Accuracy of Each Type of Attacks.

Method	**Number Features**	**ACC DOS %**	**ACC Probe %**	**ACC R2L %**	**ACC U2R %**
Proposed work	20	99.99	99.77	100.0	99.79
Eid, et.al (2010) Principle Components Analysis and Support Vector Machine based Intrusion Detection System	-	92.5	98.3	70.2	5.1

The Table 5.32 shows that this proposed work can obtain high accuracy rates for each type of attacks. Eid study applied the principle components analysis (PCA) with support vector machine (SVM).This proposed study has a higher accuracy rate for R2L attacks compared to Eid (2010), in which the accuracy rate for this study was 99% and it also obtained 70% accuracy rate. The U2R rate in this work equals 99% compared to 5.37% in Eid (2010) study.

Table 5.33 indicates that the crossover probability and the detection rate of this proposed work got higher values compared to other studies.

Table 5.33:Methods Crossover probability Detection rate

Method	**Crossover probability**	**DR%**
Proposed work	0.9	99.99
Kadam, Jadhav, 2013	0.8	91.025
Patel, Buddhadev, 2015	0.7	98.7

The Table 5.33 shows that this proposed work can obtain high detection rates with crossover probability 0.7 (41) features and when using FLAME the crossover probability 0.9 and 20 features compared to others Kadam, Jadhav, 2013study applied an effective rule generation for intrusion detection system using genetics algorithm. This proposed study has a higher detection rate 99.99% compared to Kadam, Jadhav (2013), in which the detection rate for this study was 91.025% and crossover probability 0.8 and it also obtained 98.7%%detection rate in Patel, Buddhadev, (2015) study with the crossover probability 0.7.

Table 5.34:Methods Mutation probability Detection rate

Method	Mutation probability pm	DR%
Proposed Work	0.3	99.99
Kadam, Jadhav, 2013	0.88	91.025
Patel, Buddhadev, 2015	0.01	98.7

The Table 5.34 shows that this proposed work can obtain high DR with mutation probability 0.1 (41) features and when using FLAME the mutation probability 0.3 (20) features compared to others Kadam, Jadhav, (2013) study applied an effective rule generation for intrusion detection system using genetics algorithm. This proposed study has a higher DR 99.99% compared to Kadam, Jadhav (2013), in which the DR for this study was 91.025% and mutation probability 0.088 and it also obtained 98.7%% DR in Patel, Buddhadev, (2015) study with the mutation probability 0.01.

Table 5.35:Methods Crossover probability Accuracy

Method	Crossover probability	ACC%
Proposed work	0.9	99.89
Agravat, Rao, (2011) Fuzzy Genetic-based Learning algorithms	0.9	98.5

Pawar, Bichkar, 2014 Selecting GA Parameters for Intrusion Detection	0.6	98
Danane, Parvat, 2015 fuzzy - genetic algorithm	0.8	98

The Table 5.35 shows that this proposed work can obtain high detection rates with crossover probability 0 .7 (41) features and when using FLAME the crossover probability 0.9 (20) features compared to others. Agravat and Rao, (2011) study applied a Fuzzy Genetic-based Learning algorithms This proposed study has a higher ACC 99.89% compared Agravat and Rao, (2011), in which the ACC for this study was 98.5 % and crossover probability 0.9, it also obtained 98% Accuracy in Pawar, Bichkar, (2014) study with the crossover probability 0.6 and it also obtained 98% Accuracy in Danane, Parvat, (2015) study with the crossover probability 0.8.

Table 5.36:Methods Mutation probability Accuracy

Method	**Mutation probability**	**ACC%**
Proposed work	0.3	99.89
Agravat, Rao, 2011 Fuzzy Genetic-based Learning algorithms	0.1	98.5

Pawar, Bichkar, 2014 Selecting GA Parameters for Intrusion Detection	0.1	98
Danane, Parvat, 2015 fuzzy - genetic algorithm	0.088	98

The Table 5.36 shows that this proposed work can obtain high DR with mutation probability 0 .1 (41) features and when using FLAME the mutation probability 0.3 (20) features compared to others.Agravat and Rao, (2011) study applied a Fuzzy Genetic-based Learning algorithms This proposed study has a higher ACC 99.89%compared Agravat and Rao, (2011), in which the ACC for this study was 98.5 % and mutation probability 0.1, it also obtained 98% ACC in Pawar, Bichkar, 2014study with the mutation probability 0.01 and it also obtained 98% ACC in Danane, Parvat, 2015 study with the mutation probability 0.088.

Table 5.37:Methods, FPR

Method	FAR%
Proposed work	0.0049%
Shrivastava, Jain (2011) Rough Set Theory and support vector machine	7.52%
Alsharafat, 2010 Artificial Neural Network (ANN) extended Classifier System (XCS	0.9%

Fries, Terrence P (2008) A fuzzy-genetic approach to network intrusion detection	0.2%

Table 5.37 shows that the proposed work can obtain low false positive rate, 6.2994% when using 41 features and 0.0049% when using FLAME(20 features). Shrivastava, Jain (2011) study applied Rough Set Theory and support vector machine and get 7.52% FAR. Alsharafat, (2010) study ANN and (XCS) obtain FAR 0.9%. In Fries, Terrence P (2008) study applied A fuzzy-genetic approach to network intrusion detection get low value for FAR 0.2%.

Chapter

Conclusions and Future Work

Conclusions

The main outcome of this thesis is enhancing the extended classifier system by using the FLAME algorithm in the filtration process and using cuckoo search selection in the genetic algorithm.

Experimental results presented here clearly demonstrate the following successful properties of our enhanced model of intrusion detection system on KDD99 dataset.

1. The detection rate and accuracy have increased to reach about 99.89 % and false alarms decreased to reach about 0.0029 %.
2. The features were reduced from 41 to 20 by using FLAME.

Future Work

There are many improvements that can be considered as a future work aspect; such as:

1. Firefly algorithm implementation instead of FLAME.
2. Extension of recent studies to larger instances.
3. Using different types of crossovers as a single point or two points.
4. Adapting crossover probability depending on result performance.
5. Use an NSL-KDD dataset instead of 10% KDD99.

References:

- Agravat, M., and Rao, U. P. (2011), Computer intrusion detection by two-objective fuzzy genetic algorithm, **First international conference on computer science engineering and application (CCSEA). July (15-17), Hyatt regency channai, india. Available at: http://airccj. org/CSCP/vol1/csit1226. pdf**.
- Alabsi, F., and Naoum, R. (2012), Comparison of selection methods and crossover operations using steady state genetic based intrusion detection system, **Journal of Emerging Trends in Computing and Information Sciences, 3(7), 1053-1058.**
- Alsharafat, W. (2013), Applying Artificial Neural Network and eXtended Classifier System for Network Intrusion Detection, **International Arab Journal of Information Technology (IAJIT), 10(3).**
- Anderson, James P., Computer Security Threat Monitoring and Surveillance, **James P. Anderson Co., Fort Washington, Pa., 1980.**
- Barbakh, W. A., Wu, Y., & Fyfe, C. (2009), Non-standard parameter adaptation for exploratory data analysis, **(Berlin: Springer Vol. 249).**

- ensefia, H., and Ghoualmi, N. (2011), A new approach for adaptive intrusion detection, **Computational Intelligence and Security (CIS), 2011 Seventh International Conference on (pp. 983-987). IEEE.**

- Bernadó-Mansilla, E., & Garrell-Guiu, J. M. (2003), Accuracy-based learning classifier systems: models, analysis and applications to classification tasks**. Evolutionary computation, 11(3), 209-238.**

- Bull, L., and Kovacs, T. (Eds.). (2005), Foundations of learning classifier systems, **Springer Berlin Heidelberg, (Vol. 183). Available at: https://link.springer.com/book/10.1007%2Fb100387.**

- Butz, M. V., Goldberg, D. E., Lanzi, P. L., and Sastry, K. (2007), Problem solution sustenance in XCS: Markov chain analysis of niche support distributions and the impact on computational complexity.In Butz, Goldberg, Lanzi, Sastry, **Genetic Programming and Evolvable Machines, Volume 8, Issue 1, pp 5–37.**

- Butzi, M. V., & Wilson, S. W**. ,(**2001**),** An Algorithmic Description of XCS**. Soft Computing · October 2001**

- Dam, H. H., Abbass, H. A., Lokan, C., and Yao, X. (2008), Neural-based learning classifier systems, **IEEE Transactions on Knowledge and Data Engineering, 20 (1), 26-39.**

- Dam, H. H., Abbass, H. A. and Lokan, C. (2005), DXCS: An XCS system for distributed data mining, **in H.-G. Beyer and U.-M. O'Reilly (Eds), GECCO, ACM, New York, NY, pp. 1883–1890.**

- Danane, Y., and Parvat, T. (2015), Intrusion detection system using fuzzy genetic algorithm.In Pervasive Computing (ICPC), **2015 International Conference on(8-10 Jan. 2015) (pp. 1-5). IEEE.**

- Elhamahmy, M. E., Elmahdy, H. N., and Saroit, I. A. (2010), A New Approach for Evaluating Intrusion Detection System, **CiiT International Journal of Artificial Intelligent Systems and Machine Learning, 2 (11).**

- Farid, D., Darmont, J., Harbi, N., Nguyen, H. H., & Rahman, M. Z. (2009). Adaptive network intrusion detection learning: attribute selection and classification. **In International Conference on computer systems Engineering (ICCSE),** (**Dec. 2009, Bangkok, Thailand), (p. TH60000).**

- Fries, T. P. (2008), A fuzzy-genetic approach to network intrusion detection. **Proceedings of the 10th annual conference companion on Genetic and evolutionary computation (pp. 2141-2146). ACM, Atlanta, GA, USA — July 12 - 16, 2008.**
- Fu, L., and Medico, E. (2007), FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data, **BMC Bioinformatics2007 (8:3), DOI: 10.1186/1471-2105-8-3**

© Fu and Medico; licensee BioMed Central Ltd. 2007.Published: 4 January 2007.

- Gaidhane, R., Vaidya, C. and Raghuwanshi, M. (2014), Survey: Learning Techniques for Intrusion Detection System (IDS), **International Journal of Advance Foundation and Research in Computer (IJAFRC) Volume 1, Issue 2, Feb 2014. ISSN 2348 – 4853.**
- Goldberg, D. E. and Holland, J. H. (1988), Genetic algorithms and machine learning. Machine learning, **in: Journal Machine Learning archive, volume 3, Issue 2-3, October 1988. Pages 95 – 99.**
- Guerrero, M., Castillo, O., & García, M., (2015),Cuckoo Search Algorithm via Lévy Flight with Dynamic Adaptation of Parameter Using Fuzzy Logic for Benchmark Mathematical Functions**. In Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization (pp. 555-571). Springer International Publishing.**

- Hashem, S. H. (2013), Efficiency of Svm and Pca to enhance intrusion detection system, **Journal of Asian Scientific Research, 3 (4), 381.**
- Holmes, J. H., Lanzi, P. L., Stolzmann, W. and Wilson, S. W. (2002), Learning classifier systems: New models, successful applications, **Information Processing Letters, Volume 82, Issue 1, 15 April 2002, Pages 23-30.**
- Jyothsna, V., Prasad, V. R. and Prasad, K. M. (2011), A review of anomaly based intrusion detection systems, **International Journal of Computer Applications (0975 – 8887), Volume 28– No.7, August 2011** .
- Kadam, P. U. and Jadhav, P. P. (2013), An effective rule generation for intrusion detection system using genetics algorithm, **International Journal of Science, Engineering and Technology Research (IJSETR), Volume 2, Issue 10, October 2013.**
- Kamat, S. and Karegowda, Asha G., (2014), A Brief Survey on Cuckoo Search Applications, **International Journal of Innovative Research in Computer and Communication Engineering, Volume.2, Special Issue 2, May 2014.**
- Kavitha, P., and M. Usha., (2014), ANOMALY BASED INTRUSION DETECTION IN WLAN USING DISCRIMINATION ALGORITHM COMBINED WITH NAÏV

E BAYESIAN CLASSIFIER NAÏVE BAYESIAN CLASSIFIER, **Journal of Theoretical & Applied Information Technology. 4/10/2014, Vol. 62 Issue 1, p77-84. 8p.**

- Kosheleva, O., & Kreinovich, V. (2016), Why Locating Local Optima Is Sometimes More Complicated Than Locating Global Ones, **University of Texas at El Paso 500 W. University**
- Kumar, G. (2014), Evaluation Metrics for Intrusion Detection Systems-A Study, **International Journal of Computer Science and Mobile Applications, Vol.2, Issue. 11, November 2014, pg. 11-17.**
- Kumari, S. and Shrivastava, M. (2012). A study paper on IDS attack classification using various data mining techniques**, International Journal of Advanced Computer Research, Volume-2, Number-3, Issue5, September-2012.**
- Lanzi, P. L. (2008), Learning classifier systems: then and now.In Lanzi, **Evolutionary Intelligence, March 2008, Volume 1, Issue 1, pp 63–82.**
- Mitchell Melanie, (1999), An introduction to genetic algorithms, **Cambridge, Massachusetts London, England, Fifth printing 3.**
- Moghadasian, M. and Hosseini, S. P. (2014), Binary Cuckoo Optimization Algorithm for Feature Selection in High-Dimensional Datasets**, International conference on Innovative Engineering Technologies (ICIET'2014), Dec. 28-29, Bangkok (Thailand), http://dx.doi.org/10.15242/IIE.E1214027 18.**

- Mukherjee, S., & Sharma, N. (2012), Intrusion detection using naive Bayes classifier with feature reduction. **Procedia Technology, 4, 119-128.**

- Nadi, F. and Khader, A. T. (2011), A parameter-less genetic algorithm with customized crossover and mutation operators, **GECCO '11 Proceedings of the 13th annual conference on Genetic and evolutionary computation, Pages 901-908.Dublin, Ireland- July 12 -16.**
- Paliwal, S. and Gupta, R. (2012), Denial-of-service, probing & remote to user (R2L) attack detection using genetic algorithm, **International Journal of Computer Applications, Volume 60– No.19, December 2012.**
- Panahi, M. S., Yazdani, M., & Sequerloo, A. Y. (2013), Improved Detection of Intrusion to Computer Networks using Extended Classification Systems, **Global Journal of Science, Engineering and Technology, issue 7, pp. 8-12.**
- Patel, K. and Buddhadev, B. (2015), Predictive rule discovery for network intrusion detection.In **Intelligent Distributed Computing, Springer International Publishing. (pp. 287-298).**
- Pawar S.N, Bichkar R.S, (2014), Selecting GA Parameters for Intrusion Detection **International Journal of Applied Information Systems (IJAIS). Volume 6–No.7, www.ijais.org .**
- Raut, A. S. and Singh, K. R. (2014), Anomaly based intrusion detection-a review. **International Journal on Network Security, Volume 5.**

- Rehman, R. U. (2003), Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID, **Prentice Hall Professional.** PTR Upper Saddle River, New Jersey 07458, www.phptr.com.
- Richards, R. A. (1996), Zeroth-Order Shape Optimization Utilizing a Learning Classifier System, **PhD thesis, Stanford University, ACM Digital Library**.
- Roy, S., & Chaudhuri, S. S. (2013), Cuckoo search algorithm using Lévy flight: a review. **International Journal of Modern Education and Computer Science, 5 (12), 10.**
- Sampath, P. and Prabhavathy, M. (2015), Web Page Access Prediction Using Fuzzy Clustering By Local Approximation Memberships (FLAME) Algorithm, **ARPN Journal of Engineering and Applied Sciences, Volume. 10, NO. 7, APRIL 2015.**
- Sasan,H.P.S.,&Sharma,M.(2016), Intrusion detection usingfeature selection and machine learning algohrithm with misuse detection, **International Journal of Computer Science & Information Technology (IJCSIT) Vol 8, No 1.**
- Shafi, K., Kovacs, T., Abbass, H. A. and Zhu, W. (2009), Intrusion detection with evolutionary learning classifier systems, **Journal Natural Computing, volume 8,issue (1), pages 3-27.**
- Shafi, K., Abbass, H. A. and Zhu, W. (2006), The role of early stopping and population size in XCS for intrusion detection**. Asia-Pacific Conference on Simulated Evolution and Learning (pp. 50-57). Springer Berlin Heidelberg.**

- Shrivastava, S. K. and Jain, P. (2011), Effective anomaly based intrusion detection using rough set theory and support vector machine, **International Journal of Computer Applications, volume18-No (3), pages (35-41).**
- Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., & Chan, P. K. (2011),Kdd cup 1999 data. 1999),**http://kdd. ics. uci. edu/databases/kddcup99/kddcup99. html.**
- Sigaud, O., & Wilson, S. W. (2007), Learning classifier systems: a survey**, Soft Computing(A Fusion of Foundations, Methodologies and Applications),September 2007, Volume 11, Issue 11, pp 1065–1078.**
- Tiwari, S., Roy, S. S., Charaborty, S., & Kumar, A. (2013), A novel hybrid model for network intrusion detection. In Green Computing, **Communication and Conservation of Energy (ICGCE), 2013 International Conference on (pp. 685-688). 12-14 Dec. 2013, IEEE.**
- Tzima, Fani., Psomopoulos, Fotis.,& Mitkas Pericles.(2009), Using the Grid to improve the effectiveness of Learning Classifier Systems through clustering-based initialization, **Enabling grids for E-Science (EGEE)**.

- Valian, E., Mohanna, S., & Tavakoli, S. , (2011), Improved cuckoo search algorithm for feed forward neural network training, **International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.3, July 2011.**
- Yang, X. S. and Deb, S. (2009), Cuckoo search via Lévy flights. **Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on (pp. 210-214). IEEE.**
- Yao, J., Zhao, S. and Fan, L. (2006), An enhanced support vector machine model for intrusion detection, **In: Wang GY., Peters J.F., Skowron A., Yao Y. (eds) Rough Sets and Knowledge Technology. RSKT 2006. Lecture Notes in Computer Science, vol 4062 pp 538-543. Springer, Berlin, Heidelberg,**
- Yazdani, N. M., Panahi, M. S. and Poor, E. S. (2013), Intelligent Detection of Intrusion into Databases Using Extended Classifier System (XCS), **International Journal of Electrical and Computer Engineering, Vol. 3, No. 5, October 2013, pp. 708-712.**
- Zhao, S. (2007), Intrusion Detection Using the Support Vector Machine Enhanced with a Feature-weight Kernel**, PhD Thesis. University of Regina, August 2007.**

ملخص البحث

وفقا لتكنولوجيا المعلومات والاهتمام بثورات عالم الحاسوب،اصبح لهذا العالم ملفات ومعلومات وجب حمايتها من الهجمات المختلفة بأنواعها والتي تتسبب بإفسادها وتشويهها.لذلك ظهرت العديد من الخوارزميات لزيادة مستوى الحماية ولتكشف عن جميع أنواع الهجمات.

والهدف من هذه الرساله هو بناء نظام لاكتشاف الهجمات مدعم بالخوارزمية الجينينة، خوارزمية الواقواق وخوارزمية تجمع غامض من تقريب المحلي من الأعضاء (الشعلة).وهذه الخوارزميات استخدمت في البحث من اجل زيادة نسبة اكتشاف الهجمات ونسبة الدقة، وتقليل نسبة الانذار الخاطىء.

في هذه الرسالة تم تحسين الخوارزمية الجينية باستخدام خوارزمية الواقواق في اختيار أفضل الأفراد للتزواج.وأيضا استخدم خوارزمية تجمع غامض من تقريب المحلي من الأعضاء (الشعلة) التي عملت على تقليل الميزات الموجودة في النظام من ٤١ ميزة الى ٢٠. ولقد أثر ذلك على نظام اكتشاف الهجمات في الشبكات الحاسوبية بزيادة معدل اكتشاف الهجمات ليصبح ٩٩.٩ %، وتقليل معدل الانذارالخاطىء ليصبح ٠.٠٠٤%.

Appendix

A-Results with use 41 features

Table 1: Detection Rate& Accuracy& Crossover Rate & Mutation Rate &Number of optimizations(Generation) Number of solutions(Nests) & Abandoned Probability

Detection Rate	Accuracy	Cross over Rate	Mutation Rate	Number of optimizations (Generation)	Number of solutions (Nests)	Abandoned Probability
76.17106525294996	68.5539587276549	0.9	0.1	10	50	0.1
82.34584988819766	74.11126489937783	0.9	0.1	100	50	0.1
84.82566367934878	76.34309731141393	0.9	0.1	200	50	0.1
89.0723978196026	80.16515803764226	0.3	0.2	400	50	0.1
93.28230243164012	83.95407218847603	0.1	0.1	1000	50	0.1

Table 2: Detection Rate& Accuracy& Crossover Rate & Mutation Rate &Number of optimizations(Generation) Number of solutions(Nests) & Abandoned Probability

Detection Rate	Accuracy	Cross over Rate	Muta tion Rate	Number of optimiz ations (Genera tion)	Num ber of solut ions (Nest s)	Aband oned Proba bility
82.3967547 3661123	74.1570792 6295018	0.1	0.1	100	150	0.1
93.3457109 8208109	84.0111398 8387306	0.1	0.1	1000	150	0.1

Table 3: Detection Rate& Accuracy& Crossover Rate & Mutation Rate &Number of optimizations(Generation) Number of solutions(Nests) & Abandoned Probability

Detection Rate	Accuracy	Cross over Rate	Muta tion Rate	Number of optimiz ations (Genera tion)	Num ber of solut ions (Nest s)	Aband oned Proba bility
77.3235468787696	69.5911921 9089257	0.1	0.2	10	10	0.2
79.4896449 0091558	71.5406804 1082398	0.7	0.1	20	10	0.2
80.1989404 6779699	72.1790464 2101729	0.8	0.1	30	10	0.2
81.4203930 2635756	73.2783537 2372184	0.4	0.5	50	10	0.2
82.7026340 8026536	74.4323706 7223889	0.8	0.1	100	10	0.2
84.5557517 6721572	76.1001765 9049419	0.7	0.1	150	10	0.2

86.07930840986724	77.47137756888047	0.8	0.1	200	10	0.2
86.4053460272269	77.76481142450416	0.2	0.2	250	10	0.2
87.04333367517914	78.33900030766131	0.1	0.1	300	10	0.2
82.58572640930967	74.32715376837874	0.7	0.1	350	10	0.2
88.99237366163878	80.09313629547484	0.4	0.5	400	10	0.2
89.10665229461006	80.19598706514923	0.7	0.1	500	10	0.2
93.25196523688813	83.92676871319892	0.8	0.2	1000	10	0.2

Table 4: Detection Rate& Accuracy& Crossover Rate & Mutation Rate &Number of optimizations(Generation) Number of solutions(Nests) & Abandoned Probability

Detection Rate	Accuracy	Crossover Rate	Mutation Rate	Number of optimizations (Generation)	Number of solutions (Nests)	Abandoned Probability
76.95294214382567	69.2576479294431	0.4	0.5	10	50	0.2
80.78919424502112	72.71027482051909	0.7	0.2	50	50	0.2
82.4628263549971	74.2165437194974	0.9	0.1	100	50	0.2
84.67679100186699	76.20911190168037	0.8	0.4	200	50	0.2
88.3663361437059	79.5297025293354	0.8	0.1	300	50	0.2
89.35344847040524	80.41810362336452	0.9	0.1	400	50	0.2
93.08486563435329	83.77637907091821	0.8	0.1	1000	50	0.2

Table 5: Detection Rate& Accuracy& Crossover Rate & Mutation Rate &Number of optimizations(Generation) Number of solutions(Nests) & Abandoned Probability

Detection Rate	Accuracy	Crossover Rate	Mutation Rate	Number of optimizations (Generation)	Number of solutions (Nests)	Abandoned Probability
76.96955258725104	69.27259732852595	0.1	0.2	10	50	0.3
82.15985409391783	73.94386868452604	0.9	0.1	100	50	0.3
84.71005117134138	76.23904605420724	0.1	0.1	200	50	0.3
89.18028306457157	80.26225475811441	0.2	0.2	400	50	0.3
93.70057329358232	84.33051596422385	0.7	0.1	1000	50	0.3

Detection Rate	Accuracy	False alarm Rate	Cross over Rate	Muta tion Rate	Number of optimiz ations	Num ber soluti ons (nest s)	Aband od proba bility	Num ber selec ted featu res	Inde x featu res
DOS:99.970817 25912615	99.97081725 912723	0.02918274087 3849866	0.7	0.2	10	50	0.2	18	19 13 16 41 3 4 2 15 9 14 17 18 1 12 11 33 32 27
Probe:99.56046 180515578	87.02303358 972252	0.43953819484 421786							
R2L:72.2394454 3709986	72.23944543 70982	27.7605545629 0014							
U2R:99.9982360 7955655	99.99823607 955615	0.00176392044 34516387							
SY: 92.94224014523 458	89.80788309 137603	7.05775985476 5415							
DOS: 9511150614305 7	99.95111506 143193	0.04888493856 942944	0.7	0.1	100	50	0.2	18	13 16 41 15 3 4 2 9 14 17 19 20 18 1 12 11 33 32
Probe: 98.04461397580 66	21.89514661 2564966	19.5538602419 34018							
R2L: 42.70446868598 1	32.57612603 940215	57.2955313140 19							
U2R: 99.98730503322 898	99.98730503 322633	0.01269496677 102211							

Table 6:Results with use FLAME when reduce 41 features to 18 features

SY: 85.17187568911179	58.67601519882923	14.828124310888214							
DOS:49.991629145345435	2.699863934890074	50.008370854654565	0.2	0.2	10	150	0.2	18	13 16 20 19 15 3 4 2 9 41 14 17 18 1 12 33 11 32
Probe:99.99999999999997	99.99999999999997	0.00000000000003							
R2L: 51.84036394679941	48.42277299020295	48.15963605320059							
U2R: 99.98873355889957	99.98873355889812	0.011266441100431734							
SY: 75.45518166276109	62.77784262099774	24.544818337238905							
DOS: 99.99400044036443	99.99400044036338	0.005999559635569085	0.7	0.2	100	150	0.2	18	13 16 2 3 4 19 20 15 9 14 17 41 18 1 12 32 33 11
Probe:99.94179654370511	99.61232629827805	0.05820345629489054							
R2L: 100.0	100.0	0.0							
U2R: 99.99999994921905	99.99999994921832	0.00000005078095							
SY: 99.98394923332215	99.90158167196493	0.016050766677853545							
DOS: 98.28175284366174	98.28175284366242	1.7182471563382649	0.9	0.3	1000	150	0.2	18	3 4 2 13 16 15 20 19 41 9 14 17 18 1 12 11 32 33
Probe:99.94287952240106	99.03265847288722	0.0571204775989429							
R2L: 99.99999999945459	99.99999999945463	0.00000000054541							
U2R: 99.99998553745739	99.99998553746003	0.00001446254261							
SY: 99.5561544757437	99.32859921336608	0.4438455242563073							

Table 7: Results with use FLAME when reduces 41 features to 20 features

Detection Rate	Accuracy	False alarm rate	Crossover Rate	Mutation Rate	Number of optimizations	Number of solutions (nests)	Abandod probability	Number of selected features	Index features
DOS: 99.98108257246632	99.98108257246427	0.018917427533679643	٠.٧	٠.٢	10	50	0.2	20	1 2 12 13 14 9 10 11 6 7 8 3 4 5 23 24 25 20 21 22
Probe:99.98787763742045	98.6380766127519	0.012122362579546575							
R2L: 100.0	100.0	٠.٠							
U2R: 99.99999999970848	99.99999999970848	0.00000000029152							
SY: 99.99224005239881	99.65478979623117	0.007759947601186923							
DOS: 99.99999349373618	99.99999349373464	0.0000065062638	٠.٩	٠.٣	100	50	0.2	20	4 5 1 2 3 14 15 16 11 12 13 9 10 6 7 8 24 25 26 22
Probe:99.98134485898316	99.7764128349597	0.01865514101684							
R2L: 100.0	100.0	٠.٠							
U2R: 99.99897894356474	99.79909996170952	0.00102105643526329							
SY: 99.99507932407101	99.89387657260096	0.004920675928978824							
DOS:99.97813394203983	99.97813394203986	0.021866057960167495	٠.٩	٠.٣	10	150	0.2	20	7 8 9 4 5 6 2 3 1 19 20 16 17 18 13 14 12 10 11 15
Probe:99.99999999939402	99.9999999993941	0.00000000060598							
R2L:99.99999999999997	99.99999999999997	0.00000000000003							
U2R:99.99999984343604	99.99999984343458	0.00000015656396							
SY: 99.99453344621747	99.99453344621713	0.00546655378253							
DOS:99.32230447891503	92.2794683111128	0.6776955210849707	٠.٦	٠.٢	100	150	0.2	20	32 33 34 29 30 31 27 28 24 25 26 41 40 38 39 35 36 37 1 2
Probe:96.63754913897897	95.1256827815545	3.3624508610210313							
R2L: 99.99999999802037	99.99999999802037	0.00000000197963							
U2R:99.99683516815448	99.99683516815216	0.0031648318455239632							
SY: 98.98917219601721	96.85049656470996	1.0108278039827887							
DOS: 99.54691143791901	70.91758275704537	0.4530885620809926	٠.٧	٠.١	1000	150	0.2	20	19 20 21 17 18 14 15 16 11 12 13 29 30 31 27 28 24 25 26 22
Probe:99.99458060339744	99.6806625738286	0.005419396602562188							
R2L: 100.0	100.0	٠.٠							
U2R:99.96957854655608	99.82703817778747	0.03042145344392111							
SY: 99.87776764696812	92.60632087716536	0.12223235303186897							

Table8:Results with use FLAME when reduce 41 features to 25 features

Detection Rate	Accuracy	False alarm rate	Crossover Rate	Mutation Rate	Number optimization	Number solutions	Abandod probability	Number features	Index features
DOS:99.32068716738982	89.91299248345081	0.6793128326101794	0.8	0.1	10	50	.2	25	38 25 37 5 3 4 2 28 20 33 32 35 23 24 21 7 12 27 26 40 30 39 29 31 36
Probe:99.99999531031938	99.99999531031972	0.0000046896806 2							
R2L:99.99999999993045	99.99999999930448	0.0000000006955							
U2R: 100.0	100.0	0.0							
SY:99.83017061925342	97.47824694826875	0.1698293807465 7383							
DOS: 99.95661553427595	99.75706456364343	0.0433844657240 4564	0.6	0.1	100	50	.2	25	38 25 37 5 41 28 33 32 4 2 3 35 23 24 7 21 12 20 27 26 40 30 39 29 31
Probe:99.99828469281412	99.9175439101758	0.0017153071858 757585							
R2L: 100.0	100.0	0.0							
U2R:99.99771285392545	99.99771285392632	0.0022871460745 506056							
SY:99.98815327025389	99.91808033193638	0.0118467297461 18001							
DOS:99.62147999452029	93.32473411066506	0.3785200054797 144	0.7	.3	10	150	.2	25	38 2 25 37 41 5 3 4 20 28 33 21 32 35 23 24 7 12 27 26 40 30 39 29 31
Probe:99.84692090960385	99.84692090960372	0.1530790903961 4742							
R2L: 100.0	100.0	0.0							
U2R:99.99965466398353	99.9996546639847	0.0003453360164 7							
SY:99.86701389202692	98.29282742106338	0.1329861079730 8186							

DOS:99.96889689482606	99.68590797029765	0.03110310517394055	0.8	0.2	100	150	.2	25	41 38 25 37 5 4 2 3 28 21 20 33 32 35 23 24 7 12 27 26 40 30 39 29 31
Probe:99.865160201 7582	99.8318661685192	0.13483979824179926							
R2L:99.9999999999904	99.67702866371445	0.0000000000096							
U2R:99.99544025789355	99.99544025789153	0.00455974210645 1381							
SY:99.95737433861706	99.7975607651057	0.04262566138294943							
DOS:99.99580718910482	99.99580718910435	0.00419281089517 6535	0.8	.1	1000	150	.2	25	38 25 37 5 41 28 2 3 33 32 4 20 35 21 23 24 7 12 27 26 40 31 39 29 30
Probe:99.80815772880068	87.57000854827943	0.19184227119932018							
R2L: 99.99993949891228	99.99993949891255	0.00006050108772							
U2R: 99.99998965857905	99.99998965857971	0.00001034142095							
SY: 99.95097351884921	96.89143622371901	0.04902648115078989							

Table9:Results with use FLAME when reduce 41 features to 18/20/25 features

Detection Rate	Accuracy	False alarm rate	Crossover Rate	Mutation Rate	Number optimizations	Number solutions	Abandod probability	Number selected features	Index features
DOS: 99.99400044036443	99.9940004403638	0.005999559635569085	0.7	0.2	100	150	0.2	18	13 16 2 3 4 19 20 15 9 14 17 41 18 1 12 32 33 11
Probe:99.94179654370511	99.6123262982780 5	0.05820345629489054							
R2L: 100.0	100.0	0.0							
U2R: 99.99999994921905	99.9999999492183 2	0.00000005078095							
SY: 99.98394923332215	99.9015816719649 3	0.016050766677853545							
DOS: 99.99999349373618	99.9999934937346 4	0.0000065062638	٠.٩	٠.٣	100	50	0.2	20	4 5 1 2 3 14 15 16 11 12 13 9 10 6 7 8 24 25 26 22
Probe:99.98134485898316	99.7764128349597	0.01865514101684							
R2L: 100.0	100.0	٠.٠							
U2R: 99.99897894356474	99.7990999617095 2	0.00102105643526329							
SY: 99.99507932407101	99.8938765726009 6	0.004920675928978824							
DOS: 99.95661553427595	99.7570645636434 3	0.04338446572404564	0.6	.1	100	50	0.2	25	38 25 37 5 41 28 33 32 4 2 3 35 23 24 7 21 12 20 27 26 40 30 39 29 31
Probe:99.99828469281412	99.9175439101758	0.0017153071858757585							
R2L: 100.0	100.0	0.0							
U2R:99.9977128539 2545	99.9977128539263 2	0.0022871460745506056							
SY: 99.98815327025389	99.9180803319363 8	0.011846729746118001							