Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 179109, 12 pages
http://dx.doi.org/10.1155/2014/179109

Research Article

Using Genetic Algorithm to Minimize False Alarms in Insider Threats Detection of Information Misuse in Windows Environment

Maaz Bin Ahmad,[1] Adeel Akram,[2] M. Asif,[3] and Saeed Ur-Rehman[1]

[1] *Department of Electrical and Computer Engineering, Center for Advanced Studies in Engineering (CASE), Islamabad, Pakistan*
[2] *Faculty of Software & Computer Engineering, University of Engineering & Technology (UET), Taxila, Pakistan*
[3] *Department of Electrical Engineering, Muhammad Ali Jinnah University (MAJU), Islamabad, Pakistan*

Correspondence should be addressed to Maaz Bin Ahmad; maazbinahmad@yahoo.com

Received 21 March 2014; Accepted 16 October 2014; Published 11 November 2014

Academic Editor: Wudhichai Assawinchaichote

Insider threats detection problem has always been one of the most difficult challenges for organizations and research community. Effective behavioral categorization of users plays a vital role for the success of any detection mechanisms. It also helps to reduce false alarms in case of insider threats. In order to achieve this, a fuzzy classifier has been implemented along with genetic algorithm (GA) to enhance the efficiency of a fuzzy classifier. It also enhances the functionality of all other modules to achieve better results in terms of false alarms. A scenario driven approach along with mathematical evaluation verifies the effectiveness of the modified framework. It has been tested for the enterprises having critical nature of business. Other organizations can adopt it in accordance with their specific nature of business, need, and operational processes. The results prove that accurate classification and detection of users were achieved by adopting the modified framework which in turn minimizes false alarms.

1. Introduction

The notion of insider [1–3] is a general one which may represent a current or former employee, consultant, supplier, contractor, or outsourced organization. The main features [4] of insiders are that they have privileged access to the organization and its resources and usually are trusted ones. Insiders usually know a lot more about the organization and security policies implemented there. Also they usually have more privileges and trust of the organization than external attackers. Government organizations (especially related to the defense projects) have secret information and cannot afford to rely only on the defense against attacks from outside the organization. They must also consider the danger of theft and/or sabotage from someone inside the organization. So they should be able to detect and prevent the actions of insiders to keep this information safe. Otherwise the government would have to experience the serious damage which cannot be quantified in terms of financial loss.

In this paper a modified framework is presented to detect and prevent insider threats on information system of critical nature organizations. This framework is based on our previous work [5]. Human behavior is the best indicator of human intents. So a behavior driven approach is adopted in the framework. It not only incorporates the technical measures but also some psychological indicators. The deployment is done in such a manner that there is no access to the source code for all users. Only the high level command of such organizations has access to it. It enables them to perform code reviews on it in order to find out any channels of misuse of information. The summary of the contributions of the modified framework is described below:

(i) It develops a modified insider threats risk assessment methodology, which helps in reducing false alarms problems in the previous work.

(ii) It does behavior driven classification of users using GA based fuzzy classifier which eventually helps to minimize false alarms.

(iii) It mathematically as well as practically provides a way to test and run different attacks scenarios in order to validate the framework

(iv) It improves effectiveness of most of the modules by incorporating more useful behavior and functional indicators

Section 2 of the paper describes related work. In Section 3 the methodology of modified framework is presented followed by results and discussion in Section 4. Section 5 concludes the framework.

2. Related Work

Different models like Lattice [6], BLP [7], Biba [8], and Chinese-Wall [9] have been proposed to secure the information inside the organizations. For successful detection/prevention of insiders, these require some kinds of prerequisites and strictly follow the principle of least privileges. For example, in Lattice model, it is required that resources of an organization should be categorized clearly; otherwise the model would not give good results. Attack-tree approach [10] has been used to prevent unauthorized activities of insiders. It is an effective model to monitor user activities but it ignores the user's subjectivity actions. Originator based access control [11] and propagated access control list [12] have been used to prevent unauthorized access to the information system. But as the information flows through several entities these schemes become too conservative. These also may disturb the normal work flow of the organization. Bell and la Padula [13] presented multilevel security policy, which also was a very strict one. Users were divided in the different security levels and a strict-fixed security policy was applied to them. In some cases the policy hinders normal work flow of the organization as a user may have different roles at the same time. So it necessitated the need of adoptive security policy. Role based access control RBAC [14] applied policies to users in a particular role. The principle of separation of duties was applied in it strictly. There are also some models which have been based on the safety property of a system, for example, Typed Access Matrix Model [15] and Take Grant Permission Model [16, 17]. These track a sequence of commands to check the transfer of rights problem and then deal with insider threats. Document analysis method [18] is used to detect insider threats. The documents are not only monitored individually but also are treated as a member of a group. In it there may be a possibility that one document may be a member of more than one group. So complexity as well as the false positives problem arises. In Myers et al.'s work [19] the use of logs as a detecting tool has been discussed. The emphasis was not to collect all the data for maintaining logs because doing so may increase the overall cost and may decrease the efficiency. In a high traffic network system, the traffic generated due to log management takes a considerable amount of bandwidth and memory. Combination of host-based intrusion detection scheme and network based intrusion detection scheme (HIDS NIDS) [20] has been used to detect the insider threats. Insiders may disable or can interfere with the IDS because they know the system very well. Also the nature of attacks varies a lot in the insider case and IDS approach is normally geared towards the external attacks. So only depending on the IDS output may not be so beneficial. Casola et al. [21] used fuzzy logic technique to find the weaknesses in the existing security policy implementations. Such kind of techniques may help system administrator to refine the existing security policies. Caballero et al. [22] developed some indicators to assess the performance of health related processes. These indicators helped to maintain good quality of health services as well as strong confidentiality of sensitive nature data. Herrera et al. [23] optimized the rule base of a fuzzy logic controller (FLC) by applying GA. The membership functions have already been created for FLC. Lee and Takagi [24] applied GA in order to determine the number of membership functions. Belarbi and Titel [25] used binary encoding, implemented FLC as a neural network, and used the GA to train the weights. Park et al. [26] implemented a GA to automate FLC design. Pati and Sahoo [27] used GA for FLC in order to automate the rule extraction in FPGA. Díaz et al. [28] presented the use of GA as a tuning factor for fuzzy control rules. Ireland [29] used GA in the domain of intrusion detection in order to train the dataset. Subudhi and Ranasingh [30] analyzed evolutionary schemes in order to find the optimal fuzzy controller design for robotics systems.

3. Methodology

The proposed framework is comprised of different information collection components, detection system module, risk assessment module, fuzzy classifier, and offline analysis module. The model is designed to work for windows environment. In this section only the short details of each component have been presented. For detailed reading see our previous work [5].

3.1. Risk Assessment Module. It has been observed by studying previous insider attacks that insiders usually get involved in some unusual or suspicious activities prior to launching actual attacks. So observing these kinds of activities and taking proper actions in time may help us to avoid such kinds of attacks. In this module, threat score (TS) for each employee based upon different metrics (discussed below) is computed:

$$\mathrm{TS} = F_{\mathrm{attributes}} + F_{\mathrm{behavior}} + F_{\mathrm{vulnerability}}, \quad (1)$$

where $F_{\mathrm{attributes}}$ represents the key features of the user (user capabilities: 0–15), F_{behavior} represents the behavior of that user (user intents: 0–60), $F_{\mathrm{vulnerability}}$ represents the vulnerabilities in the machine used by that user (user chances of being victimized: 0–25). So TS value for each user lies in the range 0–100. The factors contributing in $F_{\mathrm{attributes}}$ value are user's physical and electronic privileges. In calculation of F_{behavior}, the main contributors are user's competence, network operations, psychological indicators, and interaction

Table 1: Ranges of normal, medium, and high TS values for user categories.

Class	Example users	TS (normal)	TS (medium)	TS (high)
1	Network admin, system admin	TS < 50	50 ≤ TS ≤ 75	TS > 75
2	Manager, scientist	TS < 35	35 ≤ TS ≤ 50	TS > 50
3	Clerk, data entry operator	TS < 20	20 ≤ TS ≤ 35	TS > 35

with honey files. $F_{\text{vulnerability}}$ is computed by the status of firewall/antivirus, open ports, security manager presence, Windows service pack installation, and so forth. For detailed reading kindly refer to our previous work [5]. The problem with this module was that the same threshold values were set for all the users irrespective of their privileges levels. This resulted in high percentages of false alarms. In order to tackle this problem some modifications were done to this module in the modified framework as presented below.

3.1.1. Improvements in Risk Assessment Module. After deeply analyzing the results obtained from simulation [5], it was found that the reason for false alarms is the different privileges levels of users. So rather than keeping the same threshold values of TS for all the users, the users were classified in three categories. Each category is comprised of users having almost the same level of privileges. Then separate values of TS were assigned for each category to assign threat levels to users (i.e., low, medium, and high). Let's take an example. Manager is a user which belongs to class 2 with respect to the access privileges. He will be in the normal range of threat levels if TS < 35, in the medium range if 35 ≤ TS ≤ 50, and in the high range if TS > 50. The ranges of TS have been presented in Table 1 for all three classes of users.

3.2. Document Viewing Application Module. All the critical documents are presented on server in encrypted form. The documents are encrypted using an encryption algorithm which takes processor ID and motherboard ID of server as a part of the encryption key. So files cannot be decrypted on any other system even if user manages to copy them to USB device and takes these to home. When a user requests for such kind of document, then based upon his authentication, access is granted to him. During run time the document is decrypted and viewed in a control environment by using our own developed viewing application. If the user is authorized to make changes in that document then he can do it but his activity would be logged.

So in short if a user is involved in making attempts of copying the document or contents of an unauthorized (read only) document, he cannot do it. If he tries to access the unauthorized files, he will be detected and prevented. If he tries to change in the application source file or tries to transfer file to his USB, and so forth, then his activities would be logged. The application is developed in C# using the functions for encryption and computing hash. SHA-1 is used to compute hash while 3DES is used for encryption purpose.

Despite all the benefits mentioned above, there was a serious shortcoming of this module. If a user's password was misused then an unauthorized person may access the confidential documents thus breaching all the security of the framework. In order to tackle this issue the key used in encryption/decryption algorithm was modified. In addition to the part of motherboard ID and processor ID of the server as a key, a pass phrase was also accommodated as a part of the key. The authorized user enters it on the run time while accessing a confidential document from the server in order to decrypt it properly. As only authorized user would know the correct pass phrase, the unauthorized user would not be able to enter it correctly. In other words he would not be able to decrypt that document in order to access it. So misuse of user's password was also handled safely by the modified document viewing application module of the framework. The normal pdf document opened inside the document viewer application is shown in the Figure 1.

3.3. Processes Information Module. Another source of data collection is from processes running on clients' machine. At the first step the running processes are matched by name against a list of blocked processes. The process would be blocked and log is maintained if a match is found. The logs tell the name of the process and name of user. If a skillful insider renames an invalid process to a valid process name, he may run this invalid process. In order to stop this type of misuse, this module stores a growing hash values list of executable files of all allowed processes. It gets path of all the running processes, computes their hash values, and then compares those against the stored hash list. If any entry mismatches then log is maintained and that process is declared suspicious for further offline analysis.

In order to further improve this module, some modifications were done. First of all the processes were not only matched by name but with process name plus ID. This helps to remove some of the misuse cases. Additionally there are certain viruses which in spite of changes in the hash values show the previous stored values. In order to tackle the effects of viruses a file-watcher mechanism was developed which notifies the renaming, deleting, modifying, and opening actions of the file containing stored hash values. This further minimizes the chances of running illegal processes on user machine thus reducing false alarms.

3.4. Email and Web Related Information Collection Module. At the first layer emails are filtered by using information like TO, FROM subject, file name and type of attachment, CC and BCC information, and so forth. These logs are then passed through filtering rules to separate the suspicious and/or forbidden emails. For example, a filtering rule may be defined for some particular file that it can only travel inside the organization through emails. So first it compares TO, CC,

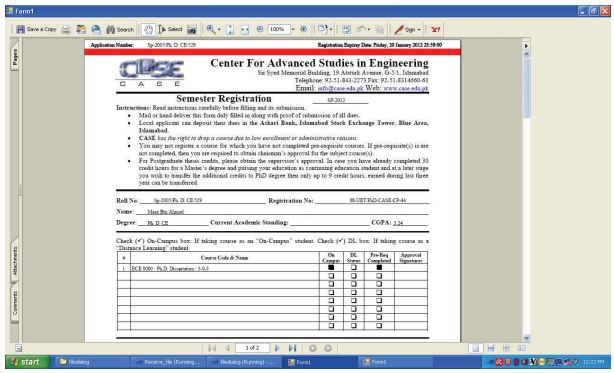

Figure 1: Viewing a normal PDF inside the Document Viewing Application.

and BCC fields to check if all of these addresses are within the organization or not. If yes then no log would be maintained. Otherwise, email is deferred for offline analysis before being sent.

There was a serious issue regarding this module. Users cannot send their legal private emails outside the organizations due to the compromise of confidentiality in the offline analysis process. So to avoid such manual inspection an application was created using a priori search algorithm of pattern matching. It reports the percentage of email contents matched with any of the confidential files. If match is found above a threshold value then it is declared suspicious. After that it is further examined in offline analysis module. So it resolved user privacy concerns to a great deal.

Data about the visited websites is also collected from web browser cache and sent to the detection system module. The logs obtained here contain email addresses of sender and recipients, attachment files, time of sending, and so forth along with trying to open forbidden sites.

3.5. OS Specific Module. Normally, users would have no access to the configuration files of Windows on their system. This module monitors changes in the important system files and configuration files which may be legal or illegal. Logs are generated if any such changes were found by the user. These changes are further analyzed in the offline analysis module to reduce any chances of false alarms. Also the time stamped logs of using system resources like printer, scanner, and so forth are also collected from event viewer after enabling auditing. This module also maintains hash values list of such files. As described earlier that in spite of changes in the file, the previous stored values are shown due to certain viruses. The same file-watcher mechanism was adopted as a second layer of defense in order to detect any kinds of changes in the files thus further minimizing false alarms.

3.6. Detection System. Data from different information sources (as described in Sections 3.2–3.5) is collected and given to the detection system. The detection system combines data and then calculates a value $\overset{\Lambda}{Y}$ (also called detection score, DS) which helps in deciding whether it is an insider attack or not:

$$\overset{\Lambda}{Y} = w1 \cdot X1 + w2 \cdot X2 + w3 \cdot X3 + w4 \cdot X4. \tag{2}$$

We have already presented the details of assigning these weights [5].

3.7. Fuzzy Classifier. As we know,

$$Y = \overset{\Lambda}{Y} + e, \tag{3}$$

where

$$e = f_N + f_P. \tag{4}$$

The error term "e" consists of false +ve and false −ve. Now the problem is to minimize false alarms. Fuzzy classifier module works as follows: note that $\overset{\Lambda}{Y}$ = DS. First inputs (i.e., TS and DS) are fuzzified. According to detection scenario, input sets (i.e., TS set and DS set) and expected outputs set (i.e., Status set) were created. First, the inputs, that is, TS and DS, were normalized. The following formula is used to normalize the inputs:

$$x_i' = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{5}$$

where $\max(x_i)$ and $\min(x_i)$ represent the minimum and maximum values of each of the attributes x_i (i.e., TS and DS). x_i' is the normalized value of these attributes (i.e., TS and DS).

After normalizing the inputs the next step is to define the linguistic variables for both inputs and outputs. In our scenario the linguistic variables are low, medium, high for inputs and attacker, suspicious and normal for outputs. The TS and DS are our inputs variables while Status is our output variable.

After this, the membership functions were created for both inputs and outputs and initial ranges were defined for these functions. It should be noted that we chose Z, triangular, and S membership functions for inputs and output. The reason of selecting these membership functions is quite obvious. As we have some knowledge about the behavior of inputs so we selected Z memberships function for the low range of both inputs and output. It is the function which keeps on decreasing from the maximum value as the value of the input increases which closely matches the characteristics of the *low* range of the inputs and output in our case. The membership function for the medium range of inputs should be such which initially starts from zero and keeps on increasing till some point. After this it should start decreasing from that point again to zero. The triangular membership function fulfills that requirement and is quite simple to implement as compared to the Gaussian function. So we selected triangular membership function for this range of inputs and outputs. Same is the reason for selecting S membership function because it keeps on increasing after some value of inputs and output till it reaches to the value of 1 and stays thereafter for all values of inputs/outputs. This was exactly what we required for the *high* range of inputs/output

values. The membership functions for inputs and out are presented as follows:

TS:

$$\mu_{\text{low(TS)}} = \begin{cases} 1; & \text{Ts} \le a \\ 1 - 2\left[\dfrac{\text{Ts} - a}{b - a}\right]^2; & a \le \text{Ts} \le \dfrac{a+b}{2} \\ 2\left[\dfrac{\text{Ts} - b}{b - a}\right]^2; & \dfrac{a+b}{2} \le \text{Ts} \le b \\ 0; & \text{Ts} \ge b \end{cases}$$

$$\mu_{\text{medium(TS)}} = \begin{cases} 0; & \text{Ts} \le c; \\ \left(\dfrac{\text{Ts} - c}{d - c}\right); & c \le \text{Ts} \le d \\ \left(\dfrac{e - \text{Ts}}{e - d}\right); & d \le \text{Ts} \le e \\ 0; & \text{Ts} \ge e \end{cases} \tag{6}$$

$$\mu_{\text{high(TS)}} = \begin{cases} 0; & \text{Ts} \le f \\ 2\left[\dfrac{\text{Ts} - f}{g - f}\right]^2; & f \le \text{Ts} \le \dfrac{f+g}{2} \\ 1 - 2\left[\dfrac{\text{Ts} - g}{g - f}\right]^2; & \dfrac{f+c}{2} \le \text{Ts} \le g \\ 1; & \text{Ts} \ge g. \end{cases}$$

DS:

$$\mu_{\text{low(DS)}} = \begin{cases} 1; & \text{Ds} \le a \\ 1 - 2\left[\dfrac{\text{Ds} - a}{b - a}\right]^2; & a \le \text{Ds} \le \dfrac{a+b}{2} \\ 2\left[\dfrac{\text{Ds} - b}{b - a}\right]^2; & \dfrac{a+b}{2} \le \text{Ds} \le b \\ 0; & \text{Ds} \ge b \end{cases}$$

$$\mu_{\text{medium(DS)}} = \begin{cases} 0; & \text{Ds} \le c; \\ \left(\dfrac{\text{Ds} - c}{d - c}\right); & c \le \text{Ds} \le d \\ \left(\dfrac{e - \text{Ds}}{e - d}\right); & d \le \text{Ds} \le e \\ 0; & \text{Ds} \ge e \end{cases} \tag{7}$$

$$\mu_{\text{high(DS)}} = \begin{cases} 0; & \text{Ds} \le f \\ 2\left[\dfrac{\text{Ds} - f}{g - f}\right]^2; & f \le \text{Ds} \le \dfrac{f+g}{2} \\ 1 - 2\left[\dfrac{\text{Ds} - g}{g - f}\right]^2; & \dfrac{f+c}{2} \le \text{Ds} \le g \\ 1; & \text{Ds} \ge g. \end{cases}$$

Status (Z):

$$\mu_{\text{low}(Z)} = \begin{cases} 1; & Z \le a \\ 1 - 2\left[\dfrac{Z - a}{b - a}\right]^2; & a \le Z \le \dfrac{a+b}{2} \\ 2\left[\dfrac{Z - b}{b - a}\right]^2; & \dfrac{a+b}{2} \le Z \le b \\ 0; & Z \ge b \end{cases}$$

$$\mu_{\text{medium}(Z)} = \begin{cases} 0; & Z \le c; \\ \left(\dfrac{Z - c}{d - c}\right); & c \le Z \le d \\ \left(\dfrac{e - Z}{e - d}\right); & d \le Z \le e \\ 0; & Z \ge e \end{cases} \tag{8}$$

$$\mu_{\text{high}(Z)} = \begin{cases} 0; & Z \le f \\ 2\left[\dfrac{Z - f}{g - f}\right]^2; & f \le Z \le \dfrac{f+g}{2} \\ 1 - 2\left[\dfrac{Z - g}{g - f}\right]^2; & \dfrac{f+c}{2} \le Z \le g \\ 1; & Z \ge g. \end{cases}$$

The initial values of the variables a, b, c, d, e, f, and g are defined below for inputs and output membership functions as shown below in the Figures 2, 3 and 4

$$\begin{aligned} a &= 0.2; \quad b = 0.4; \\ c &= 0.3; \quad d = 0.5; \\ e &= 0.7; \quad f = 0.6; \quad g = 0.8. \end{aligned} \tag{9}$$

A rule base was created (according to our knowledge about the system and inputs) through which rules could be fired according to the input values (i.e., TS and DS). The rule base remains the same as that of our previous work. Whatever combination of values of inputs is received, some rules would fire for those. Then calculation of TS and DS is performed and at the end the resulting value is defuzzified. After the defuzzification process the user is assigned to one of the output categories (i.e., attacker, suspicious, and normal) depending upon the resulting value. There are different methods available for defuzzification process; we chose "centroid" method due to its simplicity and less processing overheads.

Different attack scenarios were created [5] and tested against the system in order to find out the accuracy level of the system. The results showed that the correct classification rate was only 66.67%. This motivated us to change the membership functions and their ranges. We used genetic algorithm to tune our input membership functions ranges while keeping the output membership function constant. First we describe some concepts related to GA in order to understand it.

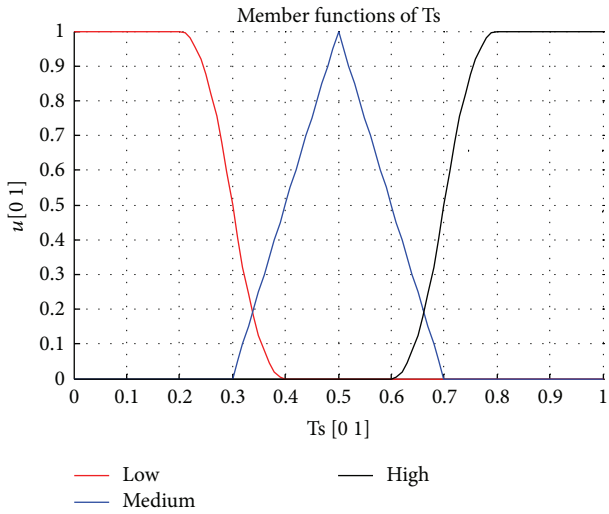

Figure 2: Initial membership functions of input TS.

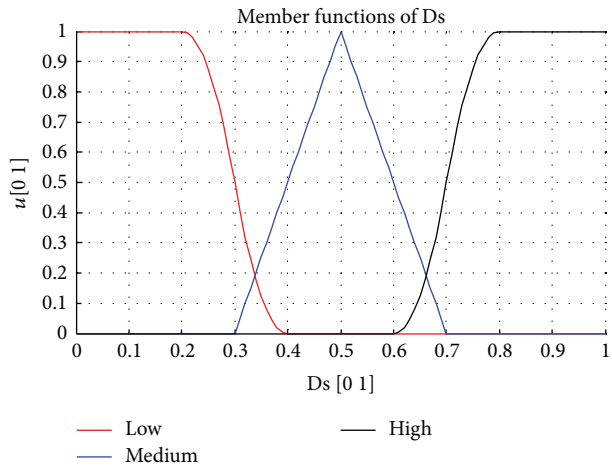

Figure 3: Membership functions of input DS.

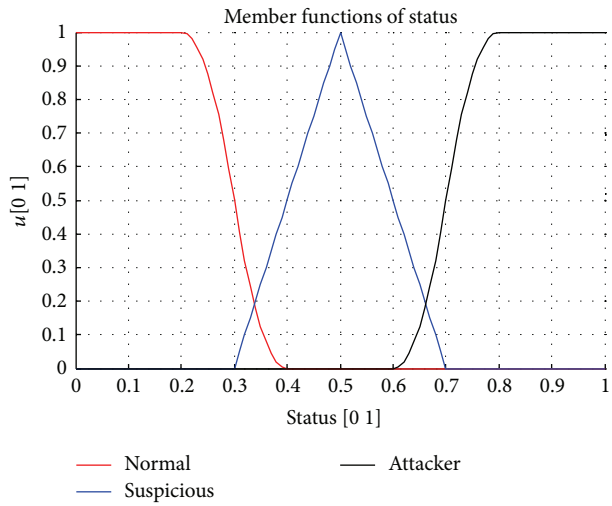

Figure 4: Membership functions of output status.

Genetic algorithms [31] are one of the robust and reliable techniques for searching the solution spaces. These algorithms are based on the concepts of evolution theory through natural selection process. The basic terminology and concepts of GA are the same as those of biological evolutionary theory. Genetic algorithms obtain a gradual improvement in the performance by simulating the biological evolution process. A population of solutions for a problem is created and a particular solution is treated as an individual in the population. The population is then allowed to evolve selecting the individuals which give the best solution to the problem. So a new population is created by trying different best combinations of individuals and with the help of the old population. It results in better solutions as the population evolves. The main motivations [32] of using GA over other traditional optimization techniques are that it is one of the most powerful optimization techniques in sampling a large solution space. GA is not much complex and is a straight forward as compared to other algorithms like particle swarm optimization (PSO). It has ability to handle multiple kinds of objectives and constraints. It is useful in situations where mathematical analysis is not available or where traditional algorithms fail. In GA the bad selection of initial parameters does not affect the final result. Some concepts related to GA are described below.

3.7.1. Chromosomes. It is a string containing values of all the parameters which need to be optimized. This string may consists of binary or real values. It is a challenging task in designing a genetic algorithm to find a suitable way of encoding the values of the parameters.

3.7.2. Population. Population is defined as a set of possible potential solutions. An individual is defined as each member of this set. The score of each individual is defined by its ability to perform better when applied to a particular problem.

3.7.3. Fitness. The score of each individual is referred to its fitness. It is the value for each individual in the population which determines the ability of an individual to solve a particular problem. This value helps in choosing the individuals from a population. There may be several ways of finding it which depend on the actual problem to be solved. For example, one way may be to run the different test cases and pass parameters of the solution through it. The output will help to determine the success level of each individual in solving the problem.

3.7.4. Mating. A mating pool is created from individuals based upon their fitness value. One particular individual may be the part of mating pool several times. It should be noted that fitness value of an individual does not guarantee that it will be a part of mating pool but its higher fitness value may increase its chances of becoming a part of the mating pool. After the creation of the mating pool, the children of the old population are calculated. The pairs are selected from the

mating pool of individuals and their solutions are combined to two new solutions. The pairs may be recombined having a chance determined by crossover rate.

3.7.5. Crossover. After selecting the pairs in the mating process, the crossover occurs among those pairs. In this process the strings of two individuals are mingled together through some method, for example, single point crossover. So new individuals are created which have properties of both of the parent individuals. Thus the process keeps on running and new individuals keep on evolving.

3.7.6. Mutation. The newly born children normally have a chance to mutate in order to maintain the genetic diversity. So, slight changes in the individuals occur in order to ensure that population may not be identical at all. Normally a slight random change is done in the chromosomes to achieve this task.

3.7.7. Next Generation. The newly created children also have fitness values which are determined. These values are compared with the old population's fitness values and the individuals from both populations having highest fitness values are chosen to develop a new generation. The size of both populations remains the same. The same process is repeated several times until a desired fitness value is achieved.

Different algorithms may be used for the above-mentioned operations. Some of the most commonly used algorithms have been presented in the following sections for different operations of the GA.

3.7.8. Selection Algorithms. There are different selections algorithms; two of those are mostly used. The first one is known as *roulette wheel* algorithm. It is basically stochastic sampling with replacement algorithm. It works in a similar way as that of roulette wheel. Every individual is given a share of the wheel. The fitness of each individual determines its share of the wheel. A random number is generated and it determines the individual to be selected. The process keeps on repeating until the desired numbers of individuals have been selected. So the fitter individuals have the greater probability to be selected due to their larger share on the wheel. But there is no guarantee that a fitter individual will always be selected for the next generation which is one of its main drawbacks.

The second algorithm is called *Stochastic Remainder Selection.* It calculates the expected occurrences of each individual selection. The resulting value is comprised of an integer portion and a floating point number for each individual. For example, the calculated value for an individual is 2.6. It means that the individual would be selected defiantly for two times and the probability of selection for the third time is 0.6. So it guarantees that all individuals having fitness level greater than that of average will be selected for the next generation.

3.7.9. Crossover Algorithms. After the selection process, the next stage is to perform the crossover. Crossover may be single point or multiple points. In single point crossover, a single point is selected across the string and after that point the strings are swapped. Multiple points are selected across strings and swapping is performed after every selected point in multipoint crossover scheme. Another crossover algorithm is *uniform crossover*. A random mask string is generated of a size the same as the parent strings. Individual bits of parents are swapped depending upon whether "0" or "1" comes in the mask string. It is the most disruptive of the crossover algorithms.

3.7.10. Mutation Algorithms. First of all a probability value (usually very low) is selected for mutation which usually remains the same throughout the GA. For mutation, a random number (in the range of 0-1) is generated for each bit. If the number is less than the selected mutation probability then that bit of the parent is flipped and vice versa. A variation to this algorithm is to use the different mutation probability for each generation starting from high to low. If the string consists of real values then one of the most commonly used algorithms for mutation is *uniform mutation*. A random value is chosen within the constraint of the variable. Also there are some other methods available like nonuniform mutation and boundary value mutation.

Table 2 describes different parameters taken for implementing the fuzzy classifier and GA. FIS type was Mamdani because it allows capturing expert knowledge in a more intuitive way [33]. The simplest and effective method of defuzzification is centroid. It is quite fast so we preferred to use it. Stochastic Remainder Selection Algorithm was selected as selection algorithm because it always guarantees that individuals having a fitness level greater than the average fitness level are always selected for the next generation. The mutation probability was taken very low as it was not actually required in our scenario. It should be noted that due to mutation and crossover there is a high risk that optimum solution may be lost [34]. So these operations need to be done with much care and should be avoided wherever not actually required. Uniform crossover algorithm was selected because it is simple having minimum risk of losing optimum solution.

Table 2: Different parameters values used in fuzzy logic and GA implementation.

Parameter	Value
FIS type	Mamdani
Defuzzification method	Centroid
Population size	100
Number of generations	100
Selection algorithm	Stochastic remainder selection
Mutation algorithm	Nonuniform
Mutation probability	0.01
Crossover algorithm	Uniform
Crossover probability	0.5

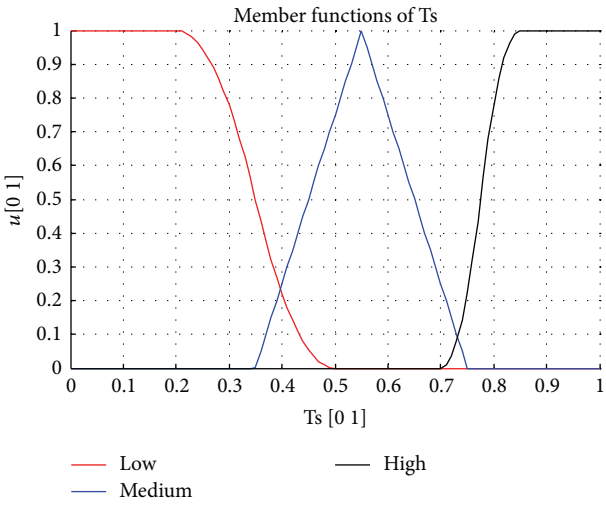

Figure 5: Modified membership functions of input TS.

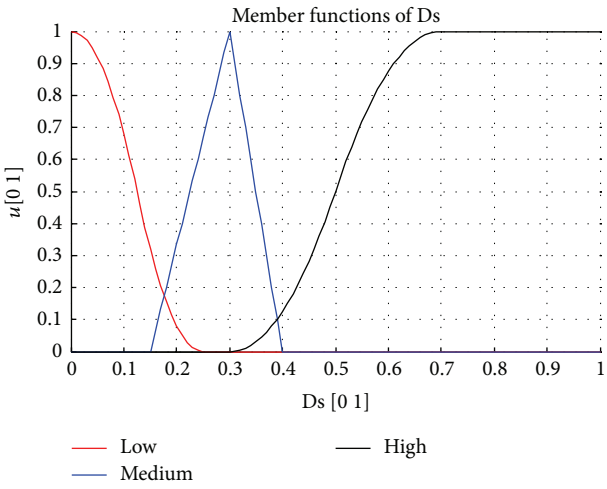

Figure 6: Modified membership functions of input DS.

The optimized membership functions after running genetic algorithm are presented in Figures 5 and 6.

Average fitness values and maximum fitness values of all iterations were computed and are presented in Figure 7.

In Figure 7, the graph shows the convergence of GA with respect to the number of iterations. As the algorithm kept on running, it started to converge and resulted in the fitness level of 1. The algorithm was configured to run for 100 iterations and it optimized the input membership functions values. The GA converged and provided the fitness values of 1. So false alarms were minimized in the fuzzy classifier module with the help of genetic algorithm.

Figure 8 describes the block diagram of the framework. It contains all modules of the framework and their relation with each other. The GA is running inside fuzzy classifier module the output of which is the category of user.

3.8. Offline Analysis Module. The offline module is very important to further refine the suspicious users' category.

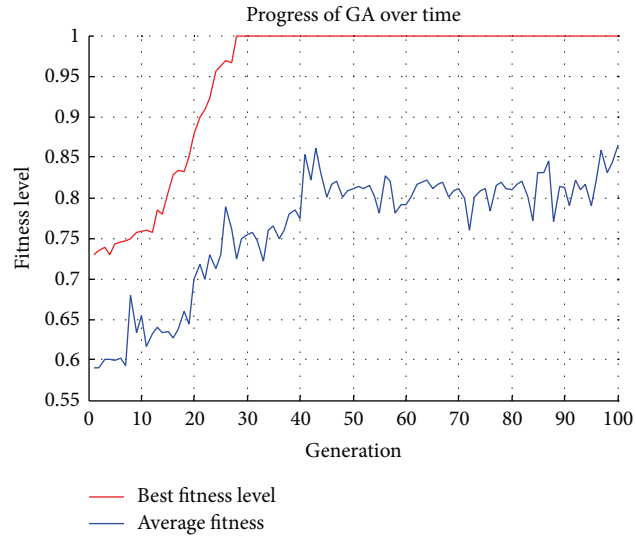

Figure 7: Average fitness values versus peak fitness values for all generations.

First it analyzes the screen shots of suspicious users. It also analyzes and compares the running processes' detail with that of peer employees' processes. In addition to these it analyzes changes in hash values of hash storing file and finds out whether it were due to some legal changes in files or not. Data about suspicious emails is also further analyzed in this module. Statistical means are used in this module to find deviations of processes' detail from normal ones. For example, CPU and memory usage of a process is compared first with its mean and max value which is maintained with the passage of time. Any abnormality results in an alarming situation and reflects the confirmation about suspicious category. Then its mean and max values are compared with the peer employees' values. If values still indicate much deviation then that user is moved to the insider category and fuzzy controller is updated. The same is done by analyzing and comparing other extra information which we have in this module for longer span of time. So this module further helps us to minimize false alarms.

In modified framework, this module also analyzes the past profile of the user in order to accurately find out the intents of the user. This module keeps record of how many times in the past the user was declared suspicious along with the reasons of declaring him so. So it refines further the user categories resulting in eliminating the false alarms and reducing the overall overhead. Its updates are sent to a fuzzy classifier to accommodate changes in the categories of users. Also a prevention based mechanism is working side by side which would not allow users of a suspicious category to damage the security of system. Note that if a user is declared as insider, then proper actions are taken against him. After that he would be under observation for one-week period at least. Then he may be randomly placed into the suspicious category with the passage of time.

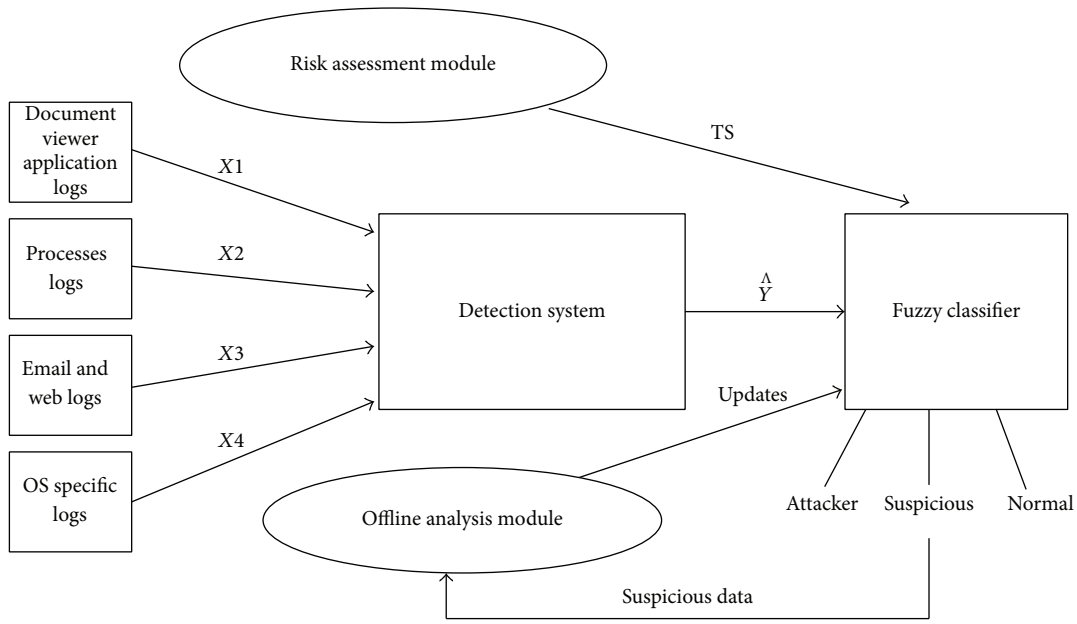

Figure 8: Insider threat detection and prevention framework.

4. Results and Discussion

We created a test bed of 4 nodes to implement the proposed framework. On server machine server 2008 OS was installed and on three client machines Windows XP, Windows Vista, and Windows 7 were installed, respectively. We installed information collection modules and risk assessment module on client and server machines and detection system, fuzzy classifier, and offline analysis modules on the server machine. On server we implemented policies regarding access to server by all clients. We designed different detailed attack scenarios [5] and tested out framework against those.

It is the organization's policy decision to select the value of threshold. Our proposed model gives us this flexibility of adjusting the threshold value. Hence it is applicable to a wide range of organizations. However we simulated our framework for an organization having critical nature of business for example defense related organizations. Several scenarios were created according to the actual working of such organizations and tested against the security offered by proposed system. These are presented and named in the format CONF[i][j][k], where CONF represents confidentiality scenario. "i" shows ith iteration and "j" represents target application number/category of scenarios (e.g., MS Word = 1 and so on). The variable "k" represents instance number/subscenarios of that scenario. As in reality the user may try to attack any time rather than attacking periodically so the framework should be capable enough to handle and detect these attacks. The simulations of scenarios were run randomly 150 times in a time of 3 hours in order to match these scenarios with real attacks and results were collected. So value of "i" will vary from 1 to 150. The value of "j" for MS Word application is 1, for PowerPoint is 2, for PDF is 3, for Excel is 4, for JPG is 5, and for TXT is 6. The values of "j" for email, processes, and super system user categories are 7, 8, and 9, respectively. The value of "k" depends on the number of instances (subscenarios) created for each targeted application/category and is numbered accordingly.

For details of these scenarios the reader may refer to our previous work [5].

In general we may simulate any scenario and can see the output of it in the form of 0 (undetected) and 1 (detected) as shown by Figure 9.

By running these scenarios, it is shown that detection rate is maximized at the low percentage of false alarms as shown in the summary of results in Table 3.

Table 3 summarizes the results of all iterations of simulated scenarios. The results prove the fact that most of these scenarios are detected very quickly in a time less than 1 second. This is so because as soon as an activity is detected which is clearly against the organization's policy it is marked as an attack. For example, suppose a user tries to use prtSc function key functionality when a confidential document is opened (e.g., as shown in [1][1][3]). This is a clear indication of his malicious desire so it will be detected very quickly. It is very rare that prtSc function key was pressed mistakenly by any user and at the same time confidential document was opened too. So false alarms ratio is also negligible (i.e., 0-1% mostly) as shown in Table 3. Whenever there is a suspicious activity detected but it is not confirmed whether it is legal or illegal, it is analyzed further in an offline analysis module. So its detection time increases also but on the other hand false alarms remain under control by doing so (e.g., as shown in [1][1][5]).

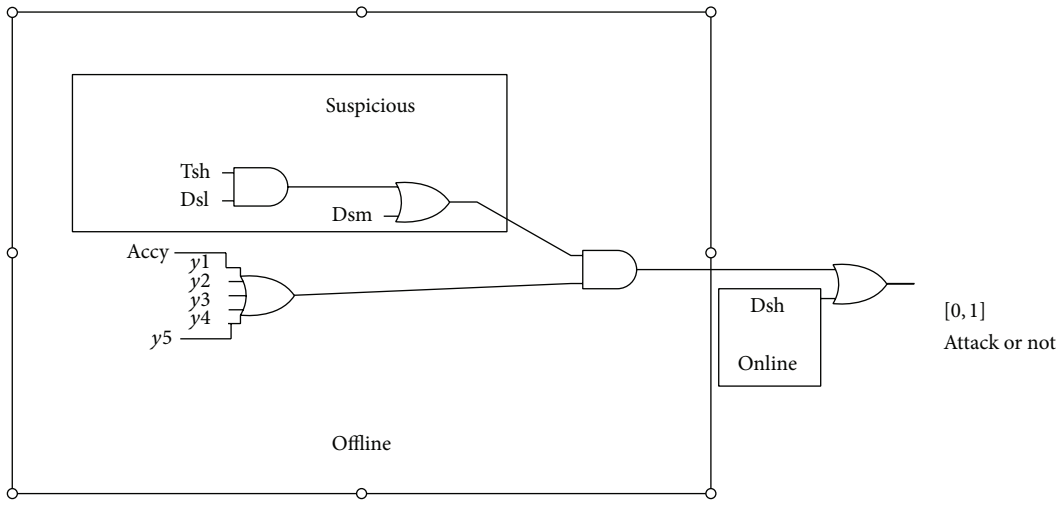

Figure 9: Validation methodology of the framework.

Table 3: Summary of simulated scenarios with respect to false alarms.

Scenario application/category	Scenario number	DS value	Initial category	Final category	False alarms (%)
MS Word	[1][1][1]-[150][1][4]	4	Attacker	Attacker	0-1
	[1][1][5]-[150][1][5]	2	Suspicious	Attacker	0–2
MS PowerPoint	[1][2][1]-[150][2][4]	4	Attacker	Attacker	0-1
	[1][2][5]-[150][2][5]	2	Suspicious	Attacker	0–2
PDF	[1][3][1]-[150][3][4]	4	Attacker	Attacker	0-1
	[1][3][5]-[150][3][5]	2	Suspicious	Attacker	0–2
MS Excel	[1][4][1]-[150][4][4]	4	Attacker	Attacker	0-1
	[1][4][5]-[150][4][5]	2	Suspicious	Attacker	0–2
JPG	[1][5][1]-[150][5][4]	4	Attacker	Attacker	0-1
	[1][5][5]-[150][5][5]	2	Suspicious	Attacker	0–2
TXT	[1][6][1]-[150][6][4]	4	Attacker	Attacker	0-1
	[1][6][5]-[150][6][5]	2	Suspicious	Attacker	0–2
Email	[1][7][1]-[150][7][2]	4	Attacker/suspicious	Attacker/normal	0-1
	[1][7][3]-[150][7][3]	2	Suspicious	Attacker/normal	0–4
	[1][7][4]-[150][7][4]	4	Attacker	Attacker/normal	0–4
Processes	[1][8][1]-[150][8][1]	4	Attacker	Attacker	0-1
	[1][8][2]-[150][8][2]	2	Suspicious	Attacker/normal	0–3
	[1][8][3]-[150][8][3]	4	Attacker	Attacker	0-1
Super system user	[1] [9][1]-[150][9][1]	2	Suspicious	Attacker/normal	0–6
	[1][9][2]-[150][9][2]	NA	None	Attacker/normal	0–8
	[1][9][3]-[150][9][3]	NA	Suspicious	Attacker/normal	0–5
	[1][9][4]-[150][9][4]	NA	None	Attacker/normal	0–8
	[1][9][5]-[150][9][5]	NA	None	None	Undefined

5. Conclusion

A modified behavior driven host based framework for insider threats detection of misuse of information has been presented in this paper. The framework upgrades most of the components of original framework in order to minimize false alarms. It also incorporates technical as well as psychological means to accomplish this task. The problem of covert channels, false alarms, and processing overheads has been tackled in such a way which does not disturb the normal legal business processes of the organization. It also keeps the higher security level at the same time. The framework has been simulated for the critical nature organization. It can also work for other types of the organizations by tweaking some of its parameters. The results obtained by simulating the attack scenarios show the effectiveness of the modified work. Most of the attacks have been detected very quickly (i.e., less than 1 sec.) while keeping the false alarms to

a minimum. More and more scenarios can be added to test the robustness and efficiency of the proposed framework. The framework provides support for different types of documents and applications/users types.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] C. Blackwell, "A security architecture to protect against the insider threat from damage, fraud and theft," in *Proceedings of the 5th Annual Cyber Security and Information Intelligence Research Workshop: Cyber Security and Information Intelligence Challenges and Strategies (CSIIRW '09)*, Oak Ridge, Tenn, USA, April 2009.

[2] M. Bishop, D. Gollmann, J. Hunker, and C. W. Probst, *Countering Insider Threats*, Dagstuhl Seminar Proceedings, 2008.

[3] A. H. Phyo and S. M. Furnell, "A detection-oriented classification of insider IT misuse," *Computers & Security Journal*, vol. 21, no. 1, 2002.

[4] T. Noonan and E. Archuleta, *The National Infrastructure Advisory's Council's Final Report and Recommendations*, 2008.

[5] B. A. Maaz, A. Adeel, U. R. Saeed, and I. Hasan, "Implementation of a behavior driven methodology for insider threats detection of misuse of information in windows environment," *Information*, vol. 16, no. 11, pp. 8121–8136, 2013.

[6] D. E. Denning, "A lattice model of secure information flow," *Communications of the Association for Computing Machinery*, vol. 19, no. 5, pp. 236–243, 1976.

[7] K. Bell and L. J. LaPadula, "Secure computer systems: unified exposition and multics interpretation," MITRE Corporation, Technical Report MTR22997, 1976.

[8] K. Biba, "Integrity considerations for secure computer systems," Tech. Rep. MTR-3153, The MITRE Corporation, 1977.

[9] D. F. C. Brewer and M. J. Nash, "The Chinese Wall security policy," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 206–214, IEEE Computer Society Press, Los Alamitos, CA, USA, May 1989.

[10] I. Ray and N. Poolsapassit, "Using attack trees to identify malicious attacks from authorized insiders," in *Computer Security-ESORICS 2005*, vol. 3679, pp. 231–246, 2005.

[11] R. Graubert, "On the need for a third form of access control," in *Proceedings of the 12th National Computer Security Conference*, pp. 296–304, October 1989.

[12] D. Wichers, D. Cook, R. Olsson et al., "PACLS: an access control list approach to anti-viral security," in *Proceedings of the 13th National Computer Security Conference*, pp. 340–349, October 1990.

[13] D. E. Bell and L. J. la Padula, "Secure computer systems. Mathematical foundations and model," Tech. Rep. M74-244, The MITRE Corporation, 1973.

[14] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role Based Access Control*, Artech House, 2003.

[15] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in operating systems," *Communications of the ACM*, vol. 19, no. 8, pp. 461–471, 1976.

[16] R. S. Sandhu, "The typed access matrix model," in *Proceedings IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 122–136, May 1992.

[17] A. Jones, R. Lipton, and L. Snyder, "A linear time algorithm for deciding security," in *Proceedings of the 17th Annual Symposium on the Foundations of Computer Science*, pp. 33–41, October 1976.

[18] H. H. Thompson, J. A. Whittaker, and M. Andrews, "Intrusion detection: perspectives on the insider threat," *Computer Fraud and Security*, vol. 2004, no. 1, pp. 13–15, 2004.

[19] J. Myers, M. R. Grimaila, and R. F. Mills, "Towards insider threat detection using web server logs," in *Proceedings of the 5th Annual Cyber Security and Information Intelligence Research Workshop: Cyber Security and Information Intelligence Challenges and Strategies (CSIIRW '09)*, Oak Ridge, Tenn, USA, April 2009.

[20] E. E. Schultz, "A framework for understanding and predicting insider attacks," *Computers and Security*, vol. 21, no. 6, pp. 526–531, 2002.

[21] V. Casola, R. Preziosi, M. Rak, and L. Troiano, "A reference model for security level evaluation: policy and fuzzy techniques," *Journal of Universal Computer Science*, vol. 11, no. 1, pp. 150–174, 2005.

[22] I. Caballero, L. E. S. Sánchez, A. Freitas, and E. Fernández-Medina, "HC+: towards a framework for improving processes in health organizations by means of security and data quality management," *Journal of Universal Computer Science*, vol. 18, no. 12, pp. 1703–1720, 2012.

[23] F. Herrera, M. Lozano, and J. L. Verdegay, "Tuning fuzzy logic controllers by genetic algorithms," *International Journal of Approximate Reasoning*, vol. 12, no. 3-4, pp. 299–315, 1995.

[24] M. A. Lee and H. Takagi, "Integrating design stages of fuzzy systems using genetic algorithms," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 612–617, San Francisco, Calif, USA, April 1993.

[25] K. Belarbi and F. Titel, "Genetic algorithm for the design of a class of fuzzy controllers: an alternative approach," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 398–405, 2000.

[26] Y. J. Park, H. S. Cho, and D. H. Cha, "Genetic algorithm-based optimization of fuzzy logic controller using characteristic parameters," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 831–836, Perth, Australia, December 1995.

[27] P. Pati and J. Sahoo, *Implementation of genetic algorithm based fuzzy logic controller with automatic rule extraction in FPGA [thesis]*, Department of Electronics & Communication Engineering National Institute of Technology, Rourkela, India, 2013.

[28] N. P. Díaz, R. L. Jiménez, and G. Angelesa, "Tuning fuzzy control rules via genetic algorithms: an experimental evaluation," *Research Journal of Recent Sciences*, vol. 2, no. 10, pp. 81–87, 2013.

[29] E. Ireland, "Intrusion detection with genetic algorithms and fuzzy logic," in *Proceedings of the UMM CSci Senior Seminar Conference*, Morris, Minn, USA, December 2013.

[30] B. Subudhi and S. Ranasingh, "Evolutionary computing approaches to optimum design of fuzzy logic controller for a flexible robot system," *Archives of Control Sciences*, vol. 23, no. 4, pp. 395–412, 2013.

[31] C. Bielza, J. A. F. del Pozo, P. Larrañaga, and E. Bengoetxea, "Multidimensional statistical analysis of the parameterization of a genetic algorithm for the optimal ordering of tables," *Expert Systems with Applications*, vol. 37, no. 1, pp. 804–815, 2010.

[32] M. Tabassum and K. Mathew, "A genetic algorithm analysis towards optimization solutions," *International Journal of Digital Information and Wireless Communications (IJDIWC)*, vol. 4, no. 1, pp. 124–142, 2014, The Society of Digital Information and Wireless Communications.

[33] S. Naaz, A. Alam, and R. Biswas, "Effect of different defuzzification methods in a fuzzy based load balancing application," *International Journal of Computer Science*, vol. 8, no. 5, pp. 261–267, 2011.

[34] P. Pati and J. Sahoo, *Implementation of genetic algorithm based fuzzy logic controller with automatic rule extraction in FPGA [Thesis]*, Department of Electronics & Communication Engineering National Institute of Technology, Rourkela, India, 2013.