A population-based incremental learning approach with artificial immune system for network intrusion detection

Meng-Hui Chen a,b, Pei-Chann Chang a,b,*, Jheng-Long Wu b

a School of Software, Nanchang University, Nanchang 330031, China
b Innovation Center for Big Data & Digital Convergence and Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan

ARTICLE INFO

Keywords:
Artificial immune system
Population-based incremental learning
Evolutionary computation
Classification problems
Electronic commerce
Collaborative filtering

ABSTRACT

The focus of this research is to develop a classifier using an artificial immune system (AIS) combined with population-based incremental learning (PBIL) and collaborative filtering (CF) for network intrusion detection. AIS is a powerful tool in terms of extirpating antigens inspired by the principles and processes of the natural immune system. PBIL uses past experiences to evolve into new species through learning and adopting the idea of CF for classification. The novelty of this research is in its combining of the three above mentioned approaches to develop a new classifier which can be applied to detect network intrusion, with incremental learning capability, by adapting the weight of key features. In addition, four mechanisms: creating a new antibody using PBIL, dynamic adjustment of feature weighting using clonal expansion, antibody hierarchy adjustment using mean affinity, as well as usage rates, are proposed to intensify AIS performance. As shown by the comparison carried out with other artificial intelligence and evolutionary computation approaches in network anomaly detection problems, our PBIL-AIS CF classifier can achieve high accuracy for the benchmark problem.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

With the advent and explosive growth of the global mobile computing and electronic commerce environments, network intrusion and anomaly detection in wide area data networks and electronic commerce infrastructures are gaining great attention from academic researchers and industrial practitioners. Recently, the number and impact of attacks has been increasing, as evidenced by recent network attacks against several prominent Web portals as well as many well-known companies. In addition, there are many types of network attacks which can damage an organization. Intrusions, such as advanced persistent threats (APTs), are designed to penetrate networks and surreptitiously steal intellectual property; distributed denial-of-service (DDoS) and flooding attacks can blind servers and shut down web sites. In addition, in the network economy, the problems caused by sustained network attacks are very difficult to handle. People worry about whether a major network disruption could cause confusion at local, national and even global levels. To detect network attacks effectively and automatically, software should be developed such that statistical signatures of network anomalies can be recognized algorithmically. This research focuses on building a classifier for network intrusion and anomaly detection in electronic commerce environments.

Classification models based on statistical pattern recognition approaches have attracted a wide range of interest in response to the growing demand for reliable and intelligent data mining systems that can be used to detect network intrusion attacks. Today, sophisticated classification is required in various domains. The goal of classification is to accurately predict the target class for each case; the determination of correct classes would lead to efficient service provision. Traditional classification systems use statistical approaches based on frequency and attack probability determination. Detection approaches such as fuzzy control, artificial neural network, decision tree, SVM, and so forth, also have good performance in numerous fault detection problems based on classification (Razavi-Far et al., 2009; Baccarini et al., 2011; Przystański and Moczulski, 2015).

In recent years, another computationally intelligent approach, Evolutionary Computing (EC) has gained wide interest in the fields of computer science and artificial intelligence. EC approaches, such as artificial immune system (AIS), genetic algorithm (GA), evolutionary strategy (ES), genetic programming (GP), ant swarm algorithm, and differential evolution (DE), have been proposed as solutions for specific anomaly detection and classification problems.
optimization (ASO) and particle swarm optimization (PSO) mimic the natural evolution mechanism to solve complex problems (Zhao, 2007; Castellani and Rowlands, 2008; Nemati et al., 2009; Agarwal et al., 2012). Hunt et al. (1998) proposed that AIS as a special approach, among these EC approaches, has a powerful evolution mechanism in accordance with the natural immune response. AIS exploits immune systems’ characteristics such as feature extraction, pattern recognition and learning and memory capabilities to intelligently modify itself with experience and continuously evolve to achieve higher accuracy. Many researchers have used AIS for parameter combination optimization (Gao, 2010; Aydin et al., 2011). Dasgupta et al. (2011) investigated the AIS approach, which effectively identifies antibodies, creates new antibodies and then incorporates Clonal Selection and Negative Selection Mechanisms. Based on their usage and mean affinity, a qualitative classification of antibodies is also maintained during the training. AIS has also been employed to solve classification problems, such as the study by Watkins et al. (2004); they developed immune inspired supervised learning algorithms and artificial immune recognition systems (AIRS). De Castro and Von Zuben (2002) proposed the computational implementation of the clonal selection principle that explicitly takes into account the affinity maturation of the immune response. Carter (2000) proposed a supervised learning system (Immunos-81) using software abstraction of T cells, B cells, antibodies and their interactions in which artificial T cells control the creation of B-cell populations (clones). However, the performances of the above AIS-based approaches are not competitive with other evolutionary or SVM-based approaches, as shown in Baccarini et al. (2011). As mentioned by Dasgupta et al. (2011), the AIS-based approach can be further improved by embedding more sophisticated mechanisms in controlling the affinity between the antibody and the antigen to improve their classification accuracy.

In this research, we incorporate population-based incremental learning (PBIL) into AIS classification to enhance the performance. Since classification is one of the most important mining tasks, we focus on developing various AIS algorithms, such as a clonal selection algorithm (CLONALG) and an antibody hierarchy adjusting mechanism. Since the existing antibodies may not be sufficient to efficiently extirpate all the antigens, the creation of new antibodies in response to the dynamic requirements becomes necessary. This can be achieved through incremental learning procedures; therefore we use PBIL to evolve new antibodies with higher affinities than the older ones, which by themselves were not capable of correctly identifying the class. The primary objective of this research is to develop a classifier by combining PBIL with AIS. In addition, the antibodies related to the target instance are clustered together using collaborative filtering for classifying the class of the target intrusions.

The rest of the paper is organized as follows. Section 2 presents a review of the literature and defines the classification problem in general. Based on the two specific problems we aim to solve in this paper, we outline a few traditional classification approaches and introduce evolutionary computation methods. Section 3 describes the methodology detailing the core algorithms used in our method. Section 4 presents the experimental results and compares our method with other classification approaches. We then close with conclusions and directions for future research.

2. Literature review

2.1. Classification problems

Classification problems require assigning items in a collection to target categories or classes. The goal is to accurately predict the target class of each case in the data set. The assignment is based on quantitative information derived from certain characteristics, such as antibody or antigen features inherent in the items as antibodies and antigens. During the classification process, a group of marked classes is provided and a training set is used to learn the definitions of these classes. Classification rules are determined, and then these rules are treated as benchmarks to identify the most likely label and the class of a new pattern (Woolley and Milanovic, 2011). Classification problems stem from various real-world situations such as those found in biological fields (Alves et al., 2010), financial services (Twala, 2010), remote sensing (Zhong et al., 2007) and inventory classification (Thomas, 2000).

Whether used to detect legitimate uses against attacks in today’s ubiquitous use of e-commerce, or the need to accurately distinguish cancer cells with the rising demands for sophisticated health, or for industrial activities such as expensive petroleum drilling operations, identification of impeccable classifiers is needed everywhere. An evaluation of classifiers thus becomes very important. Criteria such as classification precision or accuracy, the scalability of learning and classification on large data sets, the robustness to noise and the ability for incremental learning are often used to evaluate classification techniques. Scalability and incremental learning ability have become increasingly important with the collection of large amounts of data resulting from modern computing and information technologies. Since patterns embedded in large data sets generated through industrial processes may be dynamic, a classification technique should have incremental learning ability to update existing patterns with the collection of new data, and also be scalable to process data in large volumes.

An intrusion detection system (IDS) is a monitoring or a protection mechanism against various malicious activities or policy violations. With the proliferation of computers, networks and the Internet, security has become a primary concern. In general, there are two main types of IDS: network intrusion detection system (NIDS) and host-based intrusion detection system (HIDS). The intrusion detection approach usually uses statistical analysis and pattern recognition, and is capable of detecting anomaly intrusions without any prior knowledge; therefore, the model is able to generalize and extract intrusion rules during training. IDSs can reliably identify intrusion attacks in correspondence to known signatures of discovered vulnerabilities. Abadeh et al. (2011) referred to the use of genetic fuzzy systems (GFSs) in hybrid models to solve intrusion attacks problems. They presented three kinds of genetic fuzzy systems and an iterative rule learning approach to deal with intrusion detection. Altwaijry and Algarny (2012) presented a Bayesian intrusion detection system based on the Bayesian probability theory. Horng et al. (2011) presented a hierarchical clustering and support vector machines hybrid model to build an IDS. Afzali and Azmi (2014) presented a multi-agent AIS-based distributed intrusion detection system; the characteristics of MAIS-IDS are cloning, mutation, migration, collaboration and randomness. All of these researches solved the KDD Cup 1999 dataset, which is very popular in numerous studies. In this paper, we also use the KDD Cup 1999 dataset to evaluate our classification model and hence, to compare results.

Other popular classification problems also include credit card approval. With the recent growth of the credit industry, a need for an actively managed credit scoring model has emerged. A credit scoring technique is a set of decision models that assist lenders in granting consumer credit (Thomas, 2000). The models help in making decisions on whether to grant credit to new applicants based on customer characteristics such as age, income and marital status. In recent years, it has been extensively used for credit admission. The basic principle of credit scoring is to assess those who apply for fresh credit by predicting through the analyses of repayment performance on the part of previous consumers.
Initially, linear discriminate analysis was used for the evaluation even though the highly categorical data cannot be accurately assessed with linear methods. The restrictions of linearity and high use of probability distributions shattered the predictability of the credit scoring models. Since then, researchers have developed a variety of statistical models for credit scoring, such as logistic regression models (Desai et al., 1996). In recent years, many non-parametric statistical methods have been designed to solve the problem with the use of neural networks (Zhang et al., 1999) or genetic algorithms (Kim and Street, 2004).

2. Evolutionary computation

Evolutionary computation, a creative process inspired by nature's evolution, is capable of addressing complex problems with greater accuracy and speed. These problems normally involve high randomness, nonlinear dynamics and noise, which are difficult to fathom with traditional algorithms. It employs iterative progress, such as population development. The population is then guided through some parallel processing, for it to adhere to certain rules in order to achieve the desired results. The most popular EC approach is GAs, which is a family of nature inspired optimization models. The basic idea was derived from natural selection's retention of the fittest members of the population, and then carrying them forward for future selections. The fitter population leads to better output at each step and thus iteratively converges to an optimal solution. The basic concept of GA was outlined by Goldberg (1989). GAs have been applied for classification purposes at many instances in their very basic form, as well as in advanced forms. Mining classification rules from large databases (Dehuri et al., 2008) and hybrid models that combine genetic algorithms and neural networks for classifying garment defects (Yuen et al., 2009) are both examples of their applications.

In this section, we review the role of EC in classification. In computer science, artificial immune systems are based on computationally intelligent systems inspired by the principles and processes that simulate the vertebrate immune system. AIS is one of the EC methodologies that have been extensively used for developing classification approaches. It is an adaptive system, influenced by theoretical immunology and observed immune functions, principles and models, which are then applied for problem solving (Aydin et al., 2011). In addition, AIS approaches have better performance compared to artificial neural networks, GAs, fuzzy systems and so on, and have been successfully applied to many fields such as clustering, classification, pattern recognition, computer defense and optimization (Alatas and Akin, 2005; Darmoul et al., 2006; Hart and Timmis, 2008; Tavakkoli-Moghaddam et al., 2008; Aydin et al., 2010; Chang et al., 2011; Liu et al., 2012, Zhang et al., 2014).

Studies on AIS have led to enhancement in many processes, including the clonal selection (CS) mechanism and negative selection (NS) mechanism. From time to time, the NS mechanism may introduce weaker elements in the population to accommodate certain diversities. The CS mechanism captures basic features of an immune response to an antigenic stimulus, and ensures that only those antibodies that recognize antigens effectively remain in the memory cell. The CLONALG algorithm, introduced by White and Garrett (2003), primarily focuses on pattern recognition rather than explicitly on classification.

2.3. Population-based incremental learning

In this paper, we incorporate PBIL into AIS classification to enhance the performance. The concept of incremental learning involves increasing tolerance during the learning process, which tends to improve with ongoing iteration by creating new members of the population if certain threshold requirements are not met (Ye and Li, 2002). The PBIL algorithm, first proposed by Baluja and Caruana (1995), is an evolutionary optimization algorithm and estimation of distribution algorithm. It is a type of genetic algorithm (GA) whereby statistics contained in any population are explicitly maintained; the genotype of an entire population is evolved, rather than just those of the individual members (Karray and Silva, 2004). The PBIL algorithm is as follows: (1) a population is generated from a probability vector, (2) each member is evaluated and ranked based on its fitness, (3) the population probability vector is updated based on the fittest individuals, (4) mutation and (5) repeating steps 1 through 4.

PBIL has been very successful when compared against standard GAs on many benchmark and real time problems, such as the development of power system stabilizers through incremental learning (Folly, 2007); with training using incremental learning, the variation in patterns can then be used to develop models for the design of responsive stabilizers.

2.4. Collaborative filtering

A typical collaborative filtering scenario is one in which, given a set of users and a set of items, the users may rate a subset of items, while the system must predict a missing user rating for an item. Based on their information processing approaches, collaborative filtering techniques can be classified into two types: model-based and memory-based. Memory-based approaches predict the missed rating from a group of users or items with similar profiles; in this type of approach, similarity calculation is a critical step for finding a group of similar users or items.

Well-known similarity metrics include the Pearson correlation coefficient (Resnick et al., 1994), constrained Pearson (Shardanand and Maes, 1995), weighted Pearson correlation (Herlocker et al., 1999) and cosine similarity (Breese et al., 1998). Although these similarity measures have been used in many collaborative filtering algorithms, some researchers are dissatisfied with their performance. As a result, novel similarity models are continuously being put forward (Ahn, 2008; Liu et al., 2014). For example, Liu et al. (2014) proposed a similarity model taking into account the local context information of user ratings and the global preference of user behavior. They claim that this new model is more effective, especially under cold user conditions. In addition, memory-based collaborative filtering techniques achieve recommendations based on a group of similar users or items; they are also called neighborhood-based methods. Neighbor-based approaches can be further classified onto two types: user-based (Resnick et al., 1994; Shardanand, 1994) and item-based (Sarwar et al., 2001), according to the similarity of their calculation methods. User-based approaches filter information based on a group of similar users, while item-based approaches compute the similarity of items instead of user similarity.

3. Proposed approach

This study develops a classifier, named a PBIL-AISCf Classifier, to solve classification problems. The focus of this study is the generation of multiple type pools of antibodies for classification problems. Our proposed model contains three phases, as shown in Fig. 1. The first phase is the data collection and pre-processing stage; the second phase is the core application of PBIL-AISCf for evolution, which involves affinity calculation, the creation of new
impact on accuracy, sensitivity and specificity values. In essence, each element of the training data is used during the learning process and is important in shaping the final pool. The training and testing data are interested in for the datasets contain feature characteristic values. The feature value can be acquired from one of the two data types: nominal or continuous.

The nominal values use Hamming distance (HD) to compute the nominal difference \( h \). HD between two strings of follows:

\[
h_j = \sum_{j=1}^{n} w_f f_j , \quad l_j = \begin{cases} 0, & \text{if } x_{j} = x_{j}^g \\ 1, & \text{if } x_{j} \neq x_{j}^g \end{cases}
\]

where \( w_f \) represents the weight of the \( j \)th feature for the \( j \)th antibody, \( I \) represents the difference value as nominal data, \( H \) is a set of nominal data and \( x_{j}^g \) is the \( j \)th feature value of the antigen. However, we need to transform the data scale for continuous data. The normalization is done as follows:

\[
x_{j,g} = \frac{x_{j,g} - \min(x_{j})}{\max(x_{j}) - \min(x_{j})}
\]

where \( x_{j,g} \) is the normalized value for the \( j \)th antibody of the \( g \)th feature and \( x_{j,g} \) is the original feature value for the \( j \)th antibody of the \( g \)th feature. \( \min(x_{j}) \) represents the minimal value of the \( g \)th feature and \( \max(x_{j}) \) is the maximum value of the \( g \)th feature. Then, for continuous values, we use Euclidean Distance (ED) to calculate the continuous difference \( e \). The distance of continuous data as follows:

\[
e_j = \sqrt{\sum_{j=1}^{v} w_g (x_{j,g} - x_{j}^g)^2},
\]

where \( w_g \) denotes the weighting of the \( g \)th feature for the \( j \)th antibody; \( x_{j}^g \) denotes \( g \)th feature value of the antigen.

### 3.2. Evolution phase: using the PBIL-AISCF model

The evolution of PBIL-AISCF model is shown in Fig. 2.

Step 1: Initial Antibody Pool. In this step, we will generate the same \( n \) number of initial antibodies in each category pool. A random selection from the training dataset will be made. These antibodies will initiate the training process and a final exhausted antibody pool will be generated through the evolution process. We aim for sufficient information representing each class in the initial antibody pool for the AIS evolution.

Step 2: Calculating Affinity between Antibodies and Antigen. In this paper, affinity calculation is the key to what antibody can be extripated from the antigens; thus, we consider each training datum as an antigen in our system. The value for affinity is calculated for each antibody with the incoming antigen. Each antibody and antigen contains certain features. The number of features and specific characteristics defined by the features are identical for all antigens and antibodies in any given dataset. An affinity can measure the relationship of the same feature of antibody and antigen. However, the feature value has two types of data: nominal or numeric. Nominal data use HD (Eq. (1)) to obtain the different degree, whereas numeric data use ED (Eq. (3)). The affinity value for each antibody–antigen pairs is calculated as follows:

\[
Affinity_j = \frac{1}{h_j + e_j} = 1, 2, 3, ..., v.
\]
antibodies that can efficiently extirpate antigens of the right class. In our model, for an antibody to correctly extirpate an antigen, the value of affinity for the correct class of antibody has to be the best. We define the best antibody $\text{Ab}_{\text{best}}$ with the highest current affinity to form multiple antibody pools. $\text{Ab}_{\text{best}}$ is determined as follows:

$$\text{Ab}_{\text{best}} = \arg \max_{\text{Class}_k} \text{affinity}_k,$$

(5)

where Class denotes the label on classes, such as anomaly or normal.

The antibody with the best affinity for a particular antigen may not be of the extirpated antigen. This antibody hence cannot be used for detection. Nevertheless, none of the other antibodies is good enough to be used for the antigen. In order to solve this problem, we propose PBIL to create new antibody with better affinity and for correct detection.

Step 3: Create New Antibody through evolution with PBIL. As described in the previous section, the call for evolution is imperative. In this step, we seek to create a new antibody; this becomes necessary in the insufficient antibody pool during the training. The PBIL approach is employed. The antibody of the highest affinity with the particular antigen is selected from individual antibody classes. $\text{Ab}_{\text{right}}$ stands for the antibody with the highest affinity in the correct class (class of Ab and class of Ag are the same) and $\text{Ab}_{\text{wrong}}$ stands for the antibody with the highest affinity, but in the incorrect class. The evolution only becomes necessary when $\text{Affinity}_{\text{wrong}} < \text{Affinity}_{\text{right}}$. The new antibody should be identically structured as the parent antibody, and it should be generated only through a parent of the correct class.

Feature values for the new antibody $\text{Ab}_{\text{new}}$ should be calculated separately for the nominal and the continuous sets. If the feature belongs to the nominal dataset, it would be presented as shown in Eq. (6) The flowchart for the new antibody evolution is shown in Fig. 3:

$$x_{f}^{\text{Ab}_{\text{new}}} = x_{f}^{\text{Ag}}$$

(6)

$x_{f}^{\text{Ab}_{\text{new}}}$ denotes the new value of feature $f$, which is the nominal data for each $f$th nominal feature. If the feature belongs to the continuous dataset, then the concept of learning rate is used. With $LR$ being the learning rate and $N$ being the maximum number of times for evolution, according to the gradient descent algorithm, the learning rate is designated to be high initially, but tends to decrease with subsequent training. The new value of the $g$th feature in the new antibody $x_{g}^{\text{Ab}_{\text{new}}}$ can be derived as follows:

$$x_{g}^{\text{Ab}_{\text{new}}} = x_{g}^{\text{Ab}_{\text{right}}} + LR \times (x_{g}^{\text{Ab}_{\text{wrong}}} - x_{g}^{\text{Ab}_{\text{right}}}), \quad 0 < LR < 0.5.$$  

(7)

Details of the steps for the creation of a new antibody are described as follows:

1. Antibody $\text{Ab}_{\text{right}}$ and $\text{Ab}_{\text{wrong}}$ are identified using affinity rules.
2. Antibody features with high information for learning are identified. This is done by scanning the value of feature difference between the antibodies and the antigen. A value equal or close to 0 indicates close proximity, and hence is important, while higher values indicate less important features.
3. Other antibody features follow PBIL to learn.
4. Affinity of the new antibody to the antigen is re-calculated.
5. If the new antibody is of the highest affinity, then we proceed to the next step; otherwise, we go back to step 3.
6. The class of the new antibody is defined according to the antigen class.

---

Fig. 2. Flowchart for PBIL–ASS retina.

Fig. 3. The flowchart for the new antibody evolution.
7. The new antibody is injected into the antibody pool according to the antibody pool class.

The pseudo code of new antibody evolution is shown in Fig. 4.

Step 4: Clonal Expansion Mechanism. In this section, we further modify the clonal selection by introducing dynamic weight adjustment. Traditionally, only features that are important for classification are allotted higher weights, yet features distinct from the corresponding antigen features are not altered. Here, we suggest decreasing the weights of these features simply based on their distances to the corresponding features, as we believe this will further improve the affinity gap issue between the contributing antibodies and the non-contributing ones. Antibody feature weights can be adjusted from the clonal expansion rate (Eq. (8)). From Eqs. (1) and (2), \( w_f \) and \( w_g \) are derived from Eq. (7).

Let weight \( w_f \) and \( w_g \) be the weight set with a feature set \( WS = \{ w_{f1}, w_{f2}, \ldots, w_{fn} \} \), where the feature set includes nominal and numerical data:

\[
\begin{align*}
& w_{f_{new}} = w_{f_{old}} + \Delta \tau_f, \\
& \Delta \tau_f = 1 - \left( \frac{x^{g^e} - x^{f^e}}{N_g} \right),
\end{align*}
\]

where \( w_{f_{new}} \) denotes the new weight of the \( f \)th feature in the feature set; \( \Delta \tau_f \) denotes the expand rate in \( f \)th feature; \( N_g \) denotes the number of antigens. Based on formulae (8) and (9), we can expand weights of feature for learning high useful weighting using clonal expand mechanism. However, for the affinity relation, if the distance is \( A_{f_{right}} \leq A_{f_{wrong}} \), then we increase the feature weight; otherwise, if the distance is \( A_{f_{right}} \geq A_{f_{wrong}} \), then we decrease the feature weight. That said, the aim is to change or increase the relation to \( A_{f_{right}} \geq A_{f_{wrong}} \) after the clonal expansion. The flowchart and pseudo code for clonal expansion are shown in Figs. 5 and 6.

Step 5: Antibody Hierarchy Mechanism

The antibody hierarchy adjustment mechanism joins the process as the weight sets evolve, and therefore, contributes to the affinity values and usage of any antibody. The better antibody–antigen affinities and usages increases the probability of an antibody being in the memory cell, which when called upon, can dispose antigens with better accuracies during subsequent training and test runs.

The antibody hierarchy mechanism seeks to categorize antibodies in the antibody pool into “useful” and “unproductive” groups and subsequently retains the best antibodies for future training through the elimination of the unproductive ones. This step derives its importance from the fact that a less efficient antibody, if retained for testing, would lead to lower accuracy. Also a set of non-performing antibodies in the population would severely impact the accuracy parameters. At this moment, it is
desirable to keep some of the mid-range antibodies rather than
discarding them completely since they may contribute to antigen
extirpation in the future, either through a high usage rate or by
having high affinity values with a set of antigens. Thus, a cumu-
lative assessment of two parameters: Usage Rate and Mean Affinity,
is used to perform the categorization.

This mechanism divides the antibody pool into three
hierarchies:

1. Memory Cell is at the top. Most cells bear the best of the anti-
bodies, and hence are thoroughly used prior to the Mature and
Immature Cells during training. Memory cells are also used
exclusively for testing.
2. Mature Cell contains mid-range antibodies. Antibodies in this
cell may be promoted to memory cell or demoted to immature
cell depending on their performances.
3. Immature Cell contains non-performing antibodies and part of
this cell is eliminated each time the antibody hierarchy mechan-
ism is called upon during the training.

This mechanism is called upon repeatedly during training after
processing a certain number of antigens. A memory cell of high
quality is always maintained by this mechanism, and is necessary
for an efficient intrusion detection system. The constant update,
along with the mid and low range antibody classes, ensures that
minimum information is lost while eliminating the weaker anti-
bodies. The antibody and antigen match follows the following
process:

1) First Memory Cell antibodies are used to match the antigen. If
the match is successful, the new record is labeled accordingly.

   From the affinity formula, we know that the successful affinity
   relation between the two classes of antibody pool is deter-
   mined: 
   \[ \text{Affinity}_{\text{Ab}^{\text{new}}} > \text{Affinity}_{\text{Ab}^{\text{wrong}}} \].

2) If the Memory Cell antibodies are not successfully matched for
antigen, then we go to the Mature Cell to find a match.
3) Otherwise, we go to the Immature Cell. If the required condi-
tions are still not met, a need for a new antibody rises. A new
antibody \( \text{Ab}^{\text{new}} \) is to be created.

The usage rate for categorization that an antibody is capable of
detecting antigens repeatedly by following all the detection rules
has more chance of successful detection in subsequent runs and
therefore, should be kept in memory cells, and be put ahead of
those less used. This also means that it should be used prior to
those in the mature and immature cells. Since the memory cell is
called upon first for any detection, both during the training as well
as during the testing phase, exposing the best performing anti-
bodies through this cell seems logical. The usage rate is explicitly
defined for each class and each of the three cells indicated by the
rules:

\[
\text{UsageRate}_{\text{Cell}}^{\text{Class}} = \frac{T_i}{N_{\text{Cell}}^{\text{Class}}} \tag{10}
\]

where \( T_i \) stands for the total number of times the \( i \)th antibody is
successfully used during the training period. \( N_{\text{Cell}}^{\text{Class}} \) stands for
the total instances that the antibody from the cell of the particular
class was used.

Use of mean affinity for categorization. Many antibodies may
not be used frequently, but can be effective when used. Thus,
usage rate by itself would be insufficient for rating antibodies.
With this in mind, we introduce the concept of mean affinity,
which takes into account the quality of the interaction between
any antibody and antigen. Mean affinity is calculated as a ratio of
the sum of the affinities to the number of times which the
Antibody was successfully employed for antigen detection:

\[
\text{Mean Affinity} = \frac{S_i}{T_i}
\]

The sum of affinities for \( i \)th antibody during that training period is denoted by \( S_i \). \( T_i \) stands for the total number of times that \( i \)th antibody was successfully used during the training period. Fig. 7 illustrates the flow of the antibody hierarchy mechanism by depicting the combined effect of usage rate and mean affinity used for storing best antibodies in the Memory cell, while eliminating the worst ones from the immature set.

### 3.3. Classify stage of unknown instance

We have generated the classification rules for antibodies in two class pools employing PBIL-AIS\(^5\) Evolution by using only memory cell antibodies for detection since they are best in distinguishing for most cases. The use of memory cell antibodies can only be justified by the fact that the antibody hierarchy adjusting mechanism has been employed to form an effective set of antibodies through the training process. The classified strategy is based on the idea of collaborative filtering. The classification process is used to find a set of the memory cell antibodies with higher affinity to unknown instance. The affinity between the memory cell antibodies with the unknown instance is set as a threshold value, \( \alpha \). This set of memory cell antibodies with their affinities higher than the threshold value can be applied for identifying unknown intrusions related to this instance. With the classifying support of this set of memory cell antibodies, the classifier will be more powerful and accurate in network intrusions detection.

### 4. Experimental results

Data from the KDD99 Cup and the Australia Credit Approval were used to evaluate the performance of our PBIL-AIS\(^5\) integrated classification approach. We set the parameters of PBIL-AIS\(^5\) as shown in Table 1. The \( \alpha \) is set up as 0.5 for clustering a memory cell antibody to check if it is related to the target instance or not. The other parameters are set up by intensive experiments. Several traditional approaches proposed for comparisons with our approach are run in WEKA (3.6.6); their parameters are set up as follows: The cost parameter of DT (J48) is set by 0.1. The M parameter of logistic is set by 20. The gamma and cost parameters of SVM are set by \( \frac{2}{C_0} \) and 150. The affinity, clonal rate and k-nn parameter of AIRS are

```c
/* Aff: Affinity */
1) Calculate Affinities for all the antibodies with the current antigen
2) Find best antibodies in Class pools
3) Being
4) IF (Aff. (Ab\(^{\text{Normal}}\)) > Aff. (Ab\(^{\text{Anomaly}}\)) and ), then
5) Ab\(^{\text{Best}}\) = Ab\(^{\text{Normal}}\)
6) Else
7) Ab\(^{\text{Best}}\) = Ab\(^{\text{Anomaly}}\)
8) End
9) For \( i = \) number of pools
10) IF (class of Ab\(^{\text{Best}}\) in class \( i = \) Class of Antigen), then
11) Ab\(^{\text{right}}\) = Class\( _i \)
12) Else
13) Ab\(^{\text{wrong}}\) = Class\( _i \)
14) End
15) End
16) IF \( |x_f\(^{Ab}\) of Ab\(^{\text{wrong}}\) - x_f\(^{Ag}\)| > |x_f\(^{Ab}\) of Ab\(^{\text{right}}\) - x_f\(^{Ag}\)|, then
17) \( w_t\)\(^{\text{new}}\) (Ab\(^{\text{right}}\)) = \( \Delta \tau \)
18) Else IF \( |x_f\(^{Ab}\) of Ab\(^{\text{wrong}}\) - x_f\(^{Ag}\)| < |x_f\(^{Ab}\) of Ab\(^{\text{right}}\) - x_f\(^{Ag}\)|
19) \( w_t\)\(^{\text{new}}\) (Ab\(^{\text{right}}\)) = \( \Delta \tau \)
20) End
21) End
```

Fig. 6. Flowchart for the clonal expansion.
The parameters of the other approaches are set by default in WEKA. In addition, from the KDD99 Cup data, we randomly selected 49,252 records from the 10% training data, consisting of 494,021 connection records for training. We also randomly selected 31,124 records for testing. Each connection record has 41 attributes characterizing the instance, represents a sequence of packet transmission starting and ending at a time period, and can be classified as either normal connection or intrusion connection class. Each connection contains thirty-four numerical and seven categorical attributes (inputs) and one class attribute (output: normal or intrusion). As shown in the distribution of connection types in Table 2, the normal connection class has 9728 connections in the training set and 6060 connections in the testing set; the intrusion connection class has 39,524 connections in the training set and 25,064 connections in the testing set. In addition, the intrusion connection has 4 attacks: denial-of-Service (DOS), remote-to-local (R2L), user-to-root (U2R) and probing (Probe). DOS is denial of service that is accessed by

![Flowchart for antibody hierarchy adjustment.](image)

**Table 1**
Parameters of PBIL-AISPP.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>0.5</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.7</td>
</tr>
<tr>
<td>Initial antibody pool size for each class</td>
<td>20</td>
</tr>
<tr>
<td>New antibodies evolved</td>
<td>10</td>
</tr>
</tbody>
</table>

**Table 2**
Distribution of KDD99 cup data.

<table>
<thead>
<tr>
<th>Class</th>
<th>Training set</th>
<th>Testing set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal connection</td>
<td>9728</td>
<td>6060</td>
</tr>
<tr>
<td>Intrusion connection</td>
<td>39,524</td>
<td>25,064</td>
</tr>
</tbody>
</table>

**Table 3**
Distribution of KDD99 cup data.

<table>
<thead>
<tr>
<th></th>
<th>Average accuracy</th>
<th>Max accuracy</th>
<th>Min accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>97.03</td>
<td>98.21</td>
<td>95.10</td>
</tr>
<tr>
<td>Testing set</td>
<td>93.24</td>
<td>94.01</td>
<td>92.66</td>
</tr>
</tbody>
</table>

![Result on training set and testing set from PBIL-AISPP model for 10 runs.](image)
legitimate users, such as SYN flooding; R2L is unauthorized access from a remote machine or password guessing; U2R is unauthorized access to gain local super-user (root) privileges and buffer overflow attack; and Probe is surveillance and probing for information gathering.

The Australian credit approval data set, which is available publicly, has 690 samples each with 14 attributes. Each case contains six numerical and eight categorical attributes (inputs), as well as one class attribute (output: accept or reject). Fewer elements and features make the dataset suitable for faster training and testing. This dataset is interesting as there is a mixture of attributes: continuous, nominal with small numbers of values and nominal with larger numbers of values. The distribution of accepted approval class has 307 approvals, and the rejected approval class has 383 approvals for testing. In addition, we used a 10-fold cross validation mechanism in the testing stage for testing our proposed PBIL-AIS-CF classification model. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation purpose, and each observation is used for validation exactly one time.

4.1. Evaluation measures

For model evaluation, we calculated the accuracy to evaluate our proposed model. The parameters for evaluation are defined in accordance with the confusion matrix:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}
\]  

The \( TP \) denotes true positive, such as the class of classified connection and true connection are both normal in the KDD99 Cup dataset; \( TN \) denotes true negative, such as the class of classified connection and true connection are both abnormal in the KDD99 Cup dataset; \( FN \) denotes false negative such as the class of classified connection being an anomaly, but true connection is normal in the KDD99 Cup dataset and \( FP \) denotes false positive, such as the class of classified connection being normal, but true connection is an anomaly in the KDD99 Cup dataset.

4.2. Results on KDD99 cup

Performance of PBIL-AIS\(^{ST}\) classification in the sampling dataset of KDD99 Cup is shown in Table 3 and Fig. 8. The PBIL-AIS\(^{ST}\) model is average in accuracy. The maximum and minimum accuracy in 10 runs of testing set are 92.34, 94.01 and 92.66, respectively. We compared our performance against other algorithms from the literature such as PBIL-AIS without collaborative filtering, SVM, Naïve Bayes, Decision Tree (J48), k-NN, AIRS, CLONALG and Immunos-81.

Table 4 summarizes the statistics of the average accuracy values of these algorithms on the KDD99 Cup instances. To test the efficiency of our proposed PBIL-AIS\(^{ST}\) method against other evolutionary algorithms (EAs), we compared it with AIRS, CLONALG and Immunos-81. Table 3 shows that PBIL-AIS\(^{ST}\) yields a much better average accuracy ratio than the three EAs.

4.3. Results on Australian credit card

For the Australian credit card instance, we also compared the performance of our proposed algorithm with algorithms of other literature, such as PBIL-AIS without collaborative filtering, SVM, Naïve Bayes, Decision Tree (J48), k-NN, AIRS, CLONALG and Immunos-81.

Table 5 summarizes the statistics of the average accuracy values of these algorithms on the Australian credit card approval instances. To test the efficiency of our proposed PBIL-AIS\(^{ST}\) method against other EAs, we compared it with AIRS, CLONALG and Immunos-81. It can be seen from Table 5 that PBIL-AIS\(^{ST}\) yields a much better average accuracy ratio than the 9 classifiers.

5. Conclusion

We proposed a PBIL integrated AIS model for classification problems to solve for network intrusion detection and credit card validation. PBIL is a learning process that can improve the AIS evolution effect for the creation of new antibodies. The induction of PBIL strengthens the concept of retaining the fittest while eliminating weak ones in the population. In this paper, we proposed antibody hierarchy adjustment using mean affinity, as well as usage rates, to intensify AIS performance. Our proposed PBIL-AIS\(^{C\text{F}}\) classification model for classifying the KDD99 cup and Australia credit card datasets was found to yield significantly better performance. We also carried out comparisons of our method against other methods, and have shown positive performance of our method for both datasets. The major feature of our result was the efficiency with which our method works for developing IDS, as well as for designing credit scoring model. Future work can be based on further classification on various intrusion categories of the records for the KDD99 cup dataset. Our paper classified records into normal and anomaly categories only, which can be further subclassified into various kinds of anomalies. Also, implementation of other mechanisms into AIS, such as negative selection mechanism, can also improve the performance of the method.

References


