Al-Azhar University
Faculty of Science (Girls)
Department of Mathematics

Computational Intelligence in Intrusion Detection System

Thesis Submitted to Department of Mathematics, Faculty of Science, Al-Azhar University

for Obtaining the Degree of Doctor of Philosophy in

Computer Science

Submitted by

Heba Fathy Ahmed Mohamed Eid

M.Sc. in Computer Science

Lecturer Assistant

Faculty of Science, Al-Azhar University

Supervised By

Prof. Afaf Abo El Ftouh Saleh	**Prof. Aboul-Ella Hassanien**
Department of Mathematics	Department of Information Technology
Faculty of Science	Faculty of Computers & Information
Al-Azhar University.	Cairo University
	Scientific Research Group in Egypt
	(SRGE) Chairman

Cairo 2013

List of Publications

Journal Papers:

1. **Heba F. Eid** and Aboul Ella Hassanien,"Hybrid Anomaly Network Intrusion Detection: Genetic Algorithm and Hidden Nave bays Approach", KSII Transactions on Internet and Information Systems. (submited, 2013)

2. **Heba F. Eid**, Mostafa Salama and Aboul Ella Hassanien, "A Feature Selection Approach for Network Intrusion Classification: The Bi-Layer Behavioral-based", New Generation Computing . (submited, 2013)

Peer Reviewed International Conference:

1. **Heba F. Eid** , Ashraf Darwish, Aboul Ella Hassanien and Ajith Abraham, "Principle Components Analysisand Support Vector Machine Based Intrusion Detection System", The 10th IEEE international conference in Intelligent Design and Application (ISDA 2010) 29 November-1 December, Cairo Egypt, pp.363-367, 2010.

2. Mostafa Salama, **Heba F. Eid** , Ashraf Darwish and Aboul Ella Hassanien, "Hybrid Intelligent Intrusion Detection Scheme", 15th Online World Conference on Soft Computing in Industrial Applications, 15th to 27th November 2010, Springer, Advances In Intelligent and Soft Computing, pp. 295-302, 2010.

3. **Heba F. Eid**, Ashraf Darwish, Aboul Ella Hassanien and Tai-hoon Kim, "Intelligent Hybrid Anomaly Network Intrusion Detection System", International Conference on Future Generation Communication and Networking (FGIT-FGCN). CCIS/ LNCS series Springer, (Indexed by SCOPUS, EI) December 8-10, Jeju Island, Korea, pp. 209-218, 2011.

4. **Heba F. Eid**, Mostafa Salama, Aboul Ella Hassanien and Tai-Hoon Kim, "Bi-Layer Behavioral-based Feature Selection Approach for Network Intrusion Classification", The International Conference on Security Technology (FGIT-SecTech), CCIS/LNCS series Springer, (Indexed by SCOPUS, EI) December 8-10, Jeju Island, Korea, pp. 195-203, 2011.

5. **Heba F. Eid** and Aboul Ella Hassanien, "Improved Real-Time Discretize Network Intrusion Detection Model", Seventh International Conference on Bio-Inspired Computing: Theories and Application (BIC-TA), December 14-16, Gwalior, India, pp.99-109, 2012.

Abstract

Intrusion detection system (IDS) is a major research problem in network security, its goal is to dynamically identify unusual access or attacks to secure the networks. Network intrusion detection systems (NIDS) is a valuable tool for the defense in depth of computer networks. It looks for known or potential malicious activities in the network traffic and raises an alarm whenever a suspicious activity is detected. However, high-performance of NIDS pose major challenges for these systems.

One of the important research challenges for constructing high performance NIDS is dealing with data containing large number of features. Extraneous features can make it harder to detect suspicious behavior patterns, causing slow training and testing process, higher resource consumption as well as poor detection rate. Thus, feature selection (FS) is one of the key topics in NIDS. FS methods are used to reduce the dimensionality of the dataset by removing irrelevant and redundant features. FS improves the NIDS classification performance by searching for the subset of features, which improve the prediction accuracy or decrease the size of the structure without significantly decreasing prediction accuracy of the classifier built using only the selected features. Therefor, applying feature selection as a preprocessing step when building NIDS is very important if real-time detection is desired.

The thesis propose four different NIDS; PCA-SVM model, DBN-SVM model, GA-HNB model and GA-IEM-C4.5 model; such models involves data preprocessing, data reduction and intrusion classification. The proposed NID models use different intelligent algorithms and feature selection and extraction techniques. We undertake experimentation to validate our models and evaluate their performance. The experimental results on the four proposed NID models shows the models advantages of enhancing the detection accuracy and testing speed by reducing the feature dimension space. Also, it propose and validate a new feature selection approach ”Bi-Layer behavioral-based feature selection approach”, which depends on the behavior of the classification accuracy according to ranked feature. The proposed approach consists of two layers, in the first layer information gain is used to rank the features and select a new set of features. Then, in the second layer a new set of features is selected from within the first layer redacted data by searching for a group of local maximum classification accuracy. The evaluation of the proposed Bi-Layer behavioral-based feature selection approach leads to reduce the data features and improve the classification accuracy

To

My husband Ahmed Elngar, who supported me each step of the way.

My baby-girl Farida, the smile on your face brings a smile to my heart.

Acknowledgement

Firstly, I would like to thank ***ALLAH*** *for all his grace, mercy and strength that has sustained me throughout this time of my life.*

I would like to express my deep thanks and gratitude to ***Prof. Aboul Ellah Hassanien*** *for his continuous support, valuable advices and abundant experience throughout this work. He has not only provided helpful guidance but also a lot of inspiration and motivation. I have learnt many things about research working from him.*

I'm indebted to ***Prof. Afaf Abo El Ftouh Saleh****, Professor of Mathematical , Faculty of Science, Al-Azhar University for her encouragement and helpful co-operation throughout this work.*

I would like to extend my thanks to ***Dr.Mostafa Salama*** *for his sincere cooperation, support and guidance.*

Finally, I would also like to express special thanks to my family for their love and support during of this work.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Chapter 1

Introduction

In this chapter we motivate the need for securing computer systems and discuss the role of intrusion detection in their security. We give a broad overview of the field of intrusion detection as it is presented in the literature.

1.1 Motivation

Computer networks have become an essential tool of our daily life today. One of the largest networks is the internet which serves as a platform for millions of online business operation such as banking networks [1]. However, access to the Internet are now available everywhere, and at relatively low prices. This ease of accessibility introduced a new kind of criminality: **cyber-crime** [2]. Thus, network security needs to be carefully concerned to provide secure information channels.

Computer security is defined as the protection of the system against threats to confidentiality, integrity and availability [3, 4].

- **Confidentiality:** requires that network resources can only be accessed by authorized parties.

- **Integrity:** requires that network information remain unaltered by accidents or malicious attempts. i.e information can only be modified (creation, deletion and changing) by authorized parties.

- **Availability:** requires that the network resources are accessible and usable upon demand by an authorized system user.

Intrusion: Heady et al [5] describes an intrusion as any set of actions that attempt to compromise the confidentiality, integrity and availability of resources.

James P. Anderson [6] divides intruders of a computer system into four types:

1. **External intruders:** who are unauthorized users of the machines they attack of.

2. **Masquerader:** A user who gained access to the system;the masquerader can be both an external penetrator or other authorized user of the system; attempts to use the authentication information of another user.

3. **Misfeasor:** A user has legitimate access to privileged information but abuses this privilege to violate the security policy of the installation.

4. **Clandestine:** A user operates at a level below the normal auditing mechanisms, perhaps by accessing the machine with supervisory privileges.

Intrusion detection (ID) is a major research problem in network security, where the concept of ID was proposed by Anderson in 1980 [6]. ID is based on the assumption that; the behavior of intruders is different from a legal user [7].

1.2 Thesis Problem

Many research efforts have been focused on how to construct affection and accurate intrusion detection models.A variety of computational intelligence techniques have been proposed in the literature including fuzzy logic [8], neural networks [9] and support vector machines (SVM) [8, 10]. In particular, these techniques are developed to classify whether the incoming network trances are normal or intruder.

One of the important research challenges for constructing high performance IDS is dealing with data containing large number of features. IDS classification accuracy depends on the features that adequately characterize the data [11]. The amount of audit data that an IDS needs to examine is very large even for a small network.

Irrelevant and redundant features of the dataset complex IDS and reduce the detection accuracy as well. Therefor, dataset dimensional reduction is an active research area in the field of computational intelligence and pattern recognition [12, 13, 14].

In the last decade, applying feature selection techniques becoming a real prerequisite for IDS model building [15, 16, 17, 18]. The application of feature selection techniques greatly reduces the computational cost and increases the classification accuracy of classifying high dimensional data.

Several feature selection techniques have been proposed in the literature [19], [20]. Sheikhan [21] uses the chi-square technique to selected best subset of the data. Genetic algorithms (GAs) provide a simple and powerful technique for selecting subsets of features that improve the detection rates [22]. Stein et al.[23] uses a genetic algorithm to select a subset of features for decision tree classifiers. There intrusion detection model increases the detection rate and decreases the false alarm rate.

1.3 Thesis Contributions

In this thesis, we explain the need for intrusion detection system and demonstrate how to solve its above problem. The contribution of this thesis can be divided into two stages:

1. Propose, implement and test four hybrid NIDSs that combines the individual base classifiers and data preprocessing as applying feature selection algorithm and discritization. The four proposed hybrid anomaly NIDS models reduce the data features space, which leads to improve the

effectiveness and speed of the NIDS. Analysis of the proposed hybrid NIDSs with other NIDS were completed as part of this research. The results of the research demonstrated that the proposed hybride NIDSs enhance the intrusion detection rate and decreasing the testing speed.

The four proposed hybrid anomaly network IDS can be summarized as follows:

- The first proposed hybrid NIDSs **"PCA-SVM"** effectively introduced a combination of Principal Component Analysis (PCA) with Support Vector Machines (SVMs). The PCA algorithm was used in order to select a best subset of features for classifying. Then, SVM system is build to evaluate the selected subset. A series of experiments are developed on NSL-KDD dataset to examine the effectiveness of our hybrid PCA-SVM IDS. The experiment results show that the PCA-SVM IDS is able to speed up the training and testing process of intrusions detection which is important for high-speed network applications [24].
- The second proposed NIDSs **"DBN-SVM"** introduces a hybrid scheme that combines the advantages of deep belief network (DBN) and support vector machine (SVM). First, we utilize DBN to re-

duce the dimensionality of the feature sets. This is followed by SVM to classify the intrusion into five outcome; Normal, denial of service, user-to-root, remote-to-local, and probing. The proposed DBN-SVM IDS reduce the 41- dimensional of NSL-KDD dataset to approximately 87% of its original size. The DBN-SVM IDS shows higher percentage of classification than SVM and enhances the testing time due to data dimensions reduction [25].

- At the third proposed Network Intrusion Detection System (NIDS) **"GA-HNB"** we investigated the performance of Genetic algorithm-based feature selection approach to reduce the data features space and then the hidden naïve bays (HNB) approach were adapted to classify the network intrusion. In order to evaluate the performance of introduced hybrid GA-HNB IDS, several groups of experiments are conducted and demonstrated on NSL-KDD dataset. Moreover, the performances of hybrid GA-HNB IDS have been compared with the results of five well-known feature selection algorithms such as Chi square, Gain ratio and Principal component analysis (PCA). It is found that, hybrid intrusion approach produces consistently better performances on selecting the subsets of features which resulting better classification accuracies (98.63%) [26].
- Finally the fourth proposed NIDS **"IEM-GA-classifier"** a real

time discritize network ID framework is proposed. We explore the impact of applying IEM discretization and GA feature selection on the performance of network IDS. Different classifiers algorithms; rules based classifiers (Ridor, Decision table), trees classifiers (REPTree, C4.5, Random Forest) and Naïve bays classifier are used to evaluate the classification time and accuracy of the introduced network ID framework. Experiments on the NSL-KDD dataset show that IEM discretization helps to highly improve the time to classify the test instances. Which is an important factor for real time network IDS. Also, IEM discretization has a positive impact on the classification accuracy, especially for the naïve bayes classifier [27].

2. Introduce, implement and validate a new feature selection approach "Bi-Layer behavioral-based feature selection approach". The proposed approach depends on the behavior of the classification accuracy according to ranked feature. The proposed approach consists of two layers, in the first layer information gain is used to rank the features and select a new set of features depending on a global maxima classification accuracy. Then, in the second layer a new set of features is selected from within the first layer redacted data by searching for a group of

local maximum classification accuracy in order to increase the number of reduced features. The evaluation of the proposed approach leads to reduce the data features and improve the classification accuracy [28]

1.4 Thesis Organization

The thesis is organized into six chapters, as shown in Fig 1.1. **Chapter 2**; gives an overview of computational intelligence (CI). It explore the research fields of CI systems and discuss beefily different CI methods, where EC includes genetic algorithm (GA) and deep believe network (DBN). While Probabilistic methods includes hidden naïve bays (HNB) and Statistical methods includes support vector machine (SVM). Also, brief introduction of data preprocessing including data reduction and data Discretization are given. Where, two feature selection methods (principal component analysis (PCA) and The information gain (IG))are presented.

Chapter 3; illustrates the definition and taxonomy of intrusion detection system (IDS) and introduces the NSL-KDD network intrusion dataset which is used as an IDS evaluation benchmark. It Proposed four hybrid network intrusion detection models (NIDS); which enhance the classification performance and time speed of NIDS.

Chapter 4; explores how intrusion detection systems performance are evaluated. Illustrates the experimental works cases and gives the results analysis of the four proposed hybrid NIDS.

Chapter 5; First, presents the proposed feature selection approach: Bi-Layer behavioral-based feature selection. Second, gives the tow cases of the experimental work and results evaluation of the proposed approach. Case 1: Evaluate the Bi-layer behavioral-based feature selection approach; while Case 2: compares the proposed Bi-layer behavioral-based feature selection with different feature selection approaches

Finally, **Chapter 6**; concludes this thesis and lists important future work.

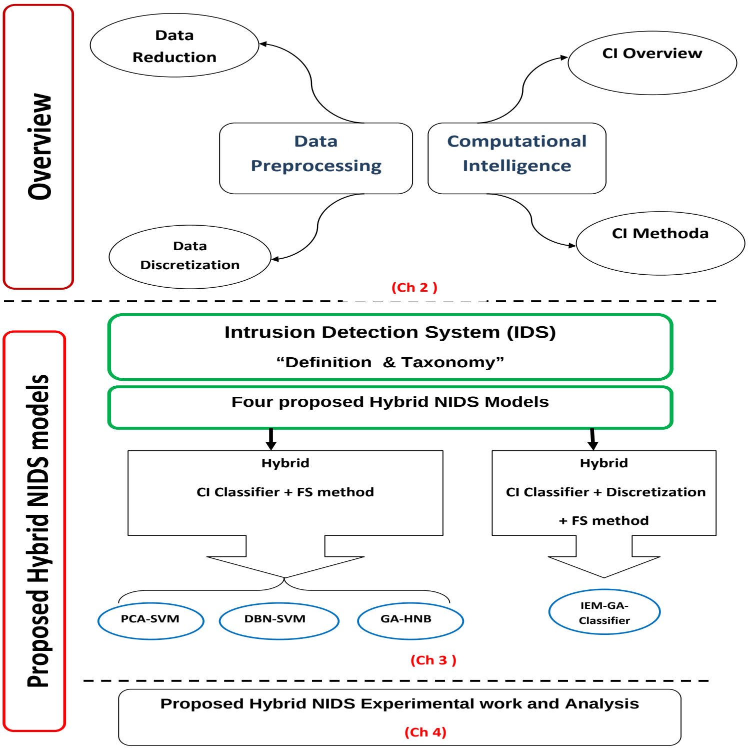

Overview
Data Reduction
CI Overview
Data Preprocessing
Computational Intelligence
Data Discretization
CI Methoda
(Ch 2)
Proposed Hybrid NIDS models
Intrusion Detection System (IDS)
"Definition & Taxonomy"
Four proposed Hybrid NIDS Models
Hybrid
CI Classifier + FS method
Hybrid
CI Classifier + Discretization
+ FS method
PCA-SVM
DBN-SVM
GA-HNB
IEM-GA-Classifier
(Ch 3)
Proposed Hybrid NIDS Experimental work and Analysis
(Ch 4)

Figure 1.1: Thesis Organization Model.

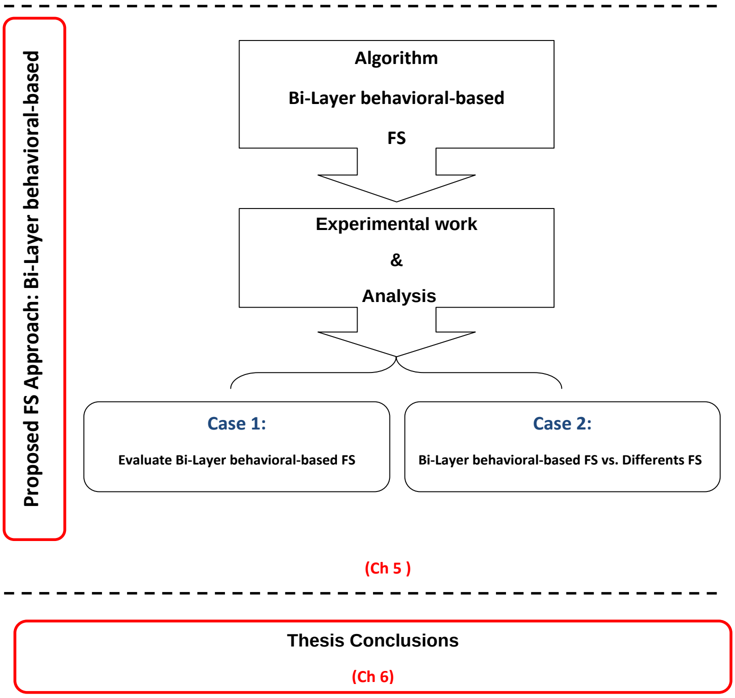

Proposed FS Approach: Bi-Layer behavioral-based
Algorithm
Bi-Layer behavioral-based
FS
Experimental work
&
Analysis
Case 1:
Evaluate Bi-Layer behavioral-based FS
Case 2:
Bi-Layer behavioral-based FS vs. Differents FS
(Ch 5)
Thesis Conclusions
(Ch 6)

Figure 1.1: Thesis Organization Model (Cont.)

CHAPTER 2

Computational Intelligence Overview

Chapter 2

Computational Intelligence Overview

2.1 Introduction

A major issue in algorithmic development is; the design of algorithmic models which are needed to solve complex problems. Enormous successes have been achieved through the modeling of biological and natural intelligence, resulting "intelligent systems (IS)“.

An intelligent system (IS) is a system that emulates some characterize of intelligence exhibited by nature. These include learning, adaptability, reasoning, and the ability to manage uncertain information [29]. Intelligent systems provide a standardized methodological to solve complex problems and obtain consistent and reliable results [30]. From the definition of IS, there is no doubt that Computational intelligence (CI) is an essential basis for building intelligent systems.

2.2 Computational Intelligence

Computational intelligence (CI) was first used in 1990 by the IEEE Neural Networks Council [31]. CI provides a combination of methods like learning, adaptation, optimization and evolution to create intelligent systems.

Attempts to find definitions of CI still bring heavy debate, many definitions of CI are proposed:

- Eberhart [32] define CI as a methodology involving computing that exhibits an ability to learn and/or deal with new situations such that the system is perceived to possess one or more attributes of reason, such as generalisation, discovery, association, and abstraction.
- The authors in [33] defined CI as: Computational Intelligence is the study of the design of intelligent agents. An intelligent agent is a system that acts intelligently: What it does is appropriate for its circumstances and its goal, it is flexible to changing environments and changing goals, it learns from experience, and it makes appropriate choices given perceptual limitations and finite computation.
- While, Bezdek [34] defined CI as: A system is computational intelligent when it: deals with only numerical (low-level) data, has pattern

recognition components, and does not use knowledge in the AI sense.

The main CI methodologies are fuzzy logic (FL), artificial neural networks (ANN) and evolutionary computation (EC)[35]. Among the CI methodologies, EC became popular in the last decade for identifying optimal solutions [36, 37, 38, 39, 40, 41]. EC is a bio-inspired search and optimization methodology based. It is capable of addressing real-world problems with great complexity [42].

In this thesis we classify the paradigms of computational intelligence into four major groups such as; Natural inspired (biological) techniques, Approximate reasoning techniques, Probabilistic techniques and Statistical techniques; Fig 2.1.

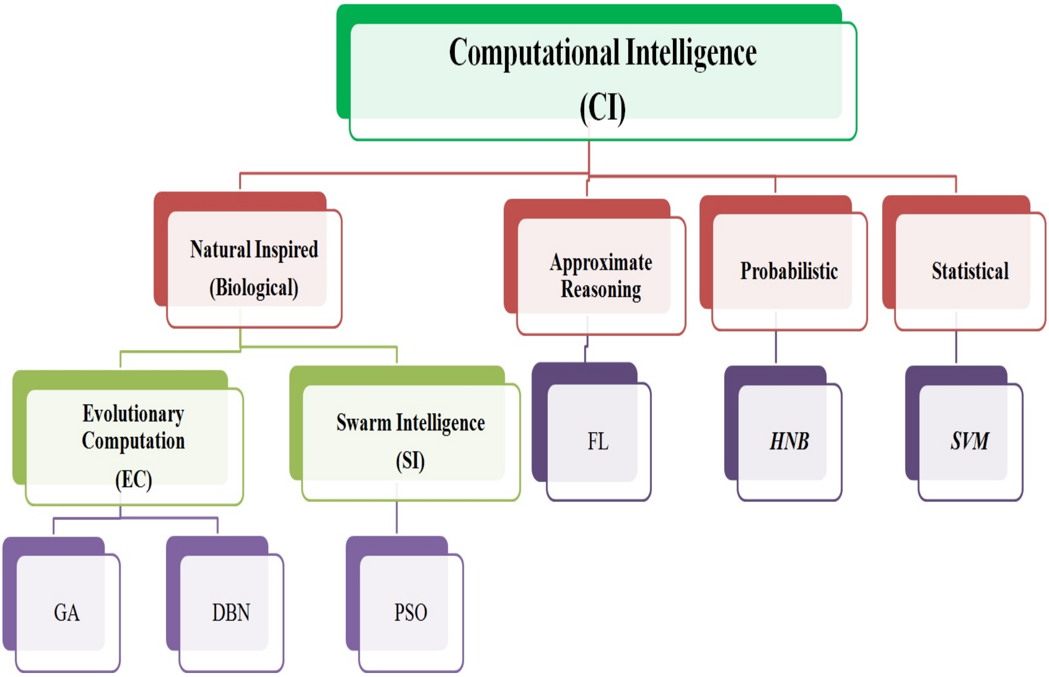


Figure 2.1: Paradigms of Computational Intelligence Research fields.

Fig 2.1 shows the research fields of CI systems, where EC includes methods like genetic algorithm (GA) and deep believe network (DBN); and swarm intelligence includes practical swarm intelligence (PSO) . While Approximate reasoning techniques contains Fuzzy logic (FL) and Probabilistic methods includes hidden naïve bays (HNB). Finally, Statistical methods includes kernel machine learning algorithms as support vector machine (SVM). These CI methods can be hybrid with feature selection methods for building fast accurate intelligent systems[8]. A brief introduction of CI methods and feature selection methods are presented in the two next section.

Computational Intelligence methods provide the frameworks of designing IS, which serves a lot of real world applications; as illustrated in Fig 2.2. This thesis is focusing on using CI methodologies as a framework of building classifiers. These classifiers are the base of designing NIDS to provide network security.

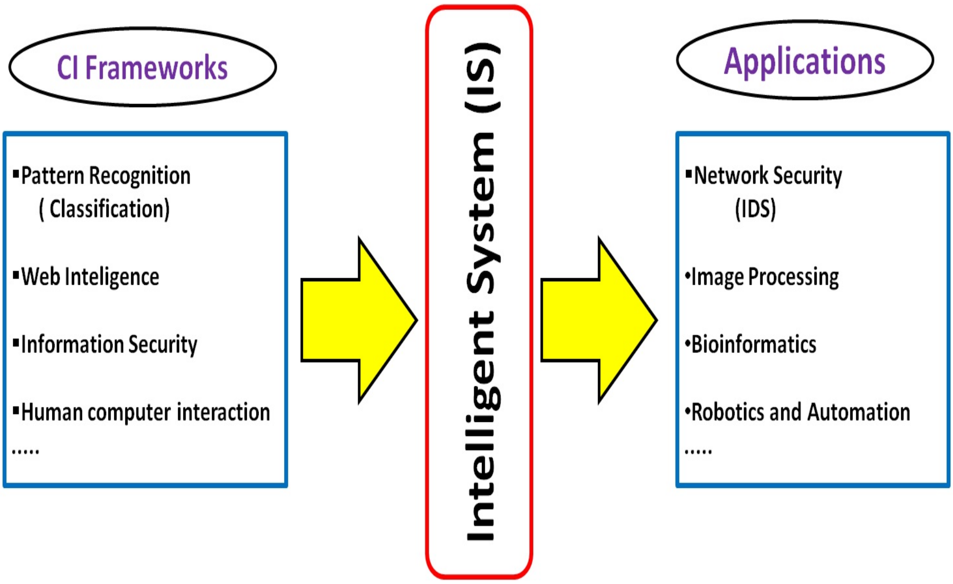


Figure 2.2: Computational Intelligence Frameworks and Real World Applications.

2.3 Computational Intelligence Methods

2.3.1 Support Vector Machines (SVM)

Support vector machine (SVM) approach is a classification technique based on Statistical Learning Theory (SLT). It is based on the idea of a hyper plane classifier, or linearly separability. The goal of SVM is to find a linear optimal hyper plane so that the margin of separation between the two classes is maximized [43, 44]. Suppose we have N training data points $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_N, y_N)\}$, where $x_i \in R^d$ and $y_i \in \{+1, -1\}$. Consider a hyper plane defined by (w, b), where w is a weight vector and b is a bias, Fig 2.3.

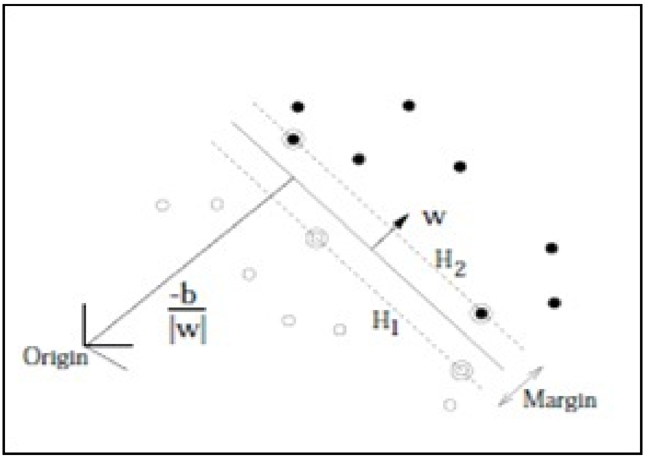


Figure 2.3: Maximum-margin hyperplane for SVM binary classification.

A new object x can be classify with the following function:

$$f(x) = sign(w.x + b) = sign(\sum_{i=1}^{N} \alpha_i y_i (x_i, x) + b) \tag{2.1}$$

where α_i are the Lagrange multipliers.

In practice, the data is often not linearly separable. However, one can still implement a linear model by transform the data points via a non-linear mapping to another higher dimensional space (feature space) such that the data points will be linear separable. This mapping is done by a kernel function K, Fig 2.4.

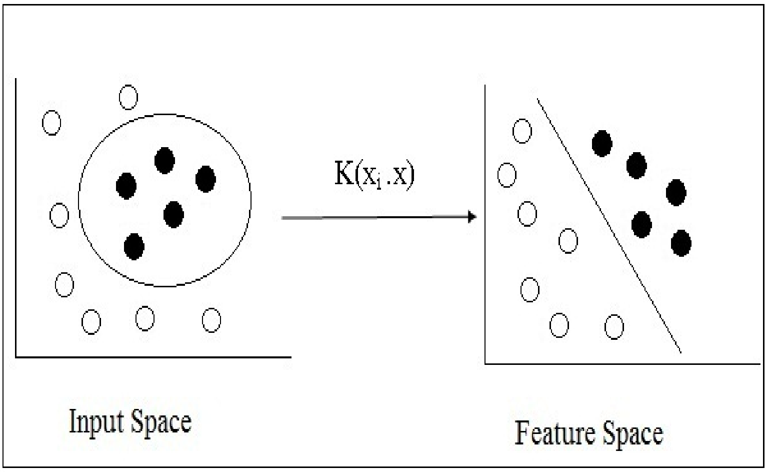


Figure 2.4: Mapping input space into feature space via kernel function.

The nonlinear decision function of SVM is given by the following function:

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b) \tag{2.2}$$

where $K(x_i, x)$ is the kernel function.

Compared with conventional machine learning methods SVMs have some advantages [45, 46]:

1. There are only two free parameters to be chosen, namely the upper bound and the kernel parameter.
2. The solution of SVM is unique, optimal and global since the training of a SVM is done by solving a linearly constrained quadratic problem.
3. Good generalization performance and Good robustness. Because of the above advantages, SVM has been recently used in many applications.

The algorithm for SVM is given in Algorithm 1.

Algorithm 1 Support Vector Machine Algorithm

```
Input:
a set D of training examples.
ker: kernel function.
Output:
Support Vectors.
1: Construct the Kernel matrix.
2: Set up the parameters for the Optimization problem.
3: Solve the Optimization Problem.
4: Compute the number of Support Vector.
5: Return the Support Vectors.
```

2.3.2 Hidden naïve bays (HNB)

Jiang et al [47] proposed hidden naïve Bayes (HNB). HNB inherits its structural from naive Bayes. It creates a hidden parent for each attribute to combine the influences from all other attributes. Hidden parents are defined by the average of weighted one-dependence estimators. The structure of HNB is given in Fig 2.5.

From Fig 2.5 $A_1, A_2, ..., A_n$ are n attributes. An instance E is represented by a vector $< a_1, a_2, ..., a_n >$, where a_i is the value of A_i. C is the class variable and c(E)represent the class of E.

In HNB each attribute A_i has a hidden parent A_{hpi} , $i = 1, 2, ..., n$,

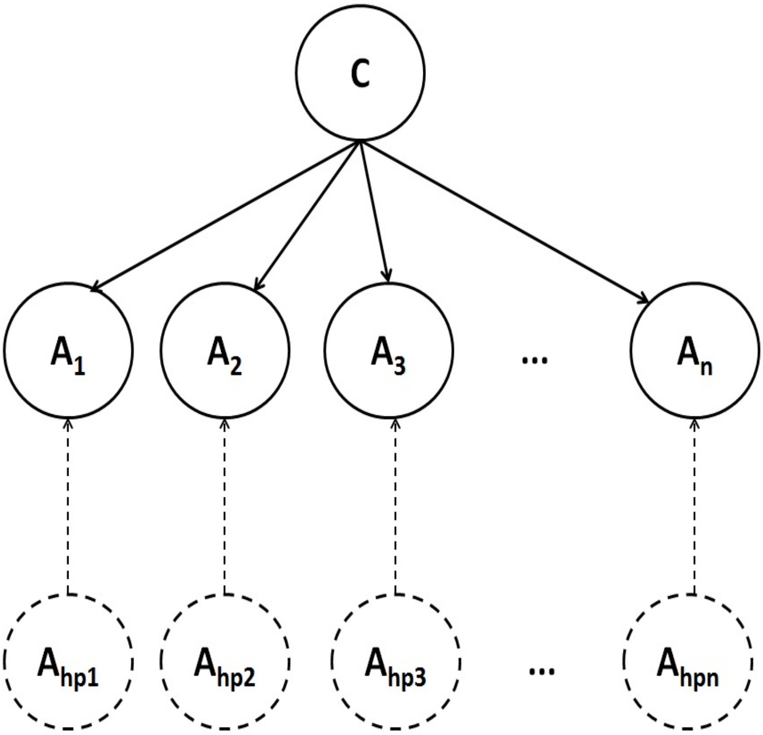


Figure 2.5: Hidden Naïve Bayes Structure.

The joint distribution represented by an HNB is given by:

$$P(A_1, ..., A_n, C) = P(C) \prod_{i=1} P(A_i \mid A_{hpi}, C) \tag{2.3}$$

where

$$P(A_i \mid A_{hpi}, C) = \sum_{j=1, j \neq i}^{n} W_{ij} * P(A_i \mid A_j, C) \tag{2.4}$$

and

$$\sum_{j=1,j\neq i}^{n} W_{ij} = 1 \tag{2.5}$$

the weight W_{ij} is compute directly from the conditional mutual information between two attributes A_i and A_j

$$W_{ij} = \frac{I_p(A_i; A_j \mid C)}{\sum_{j=1,j\neq i}^{n} I_p(A_i; A_j \mid C)} \tag{2.6}$$

where

$$I_p(A_i; A_j \mid C) = \sum_{a_i,a_j,c} P(a_i, a_j, c) log \frac{P(a_i, a_j \mid c)}{P(a_i \mid c)P(a_j \mid c)} \tag{2.7}$$

The HNB classifier on $E = (a_1, ..., a_n)$ is define as follows

$$c(E) = arg \max_{c\in C} P(c) \prod_{i=1}^{n} P(a_i \mid a_{hpi}, c) \tag{2.8}$$

The learning algorithm for HNB is given in Algorithm 2.

Algorithm 2 Hidden Naïve Bayes Algorithm

```
Input: a set D of training examples.
Output: an hidden naive Bayes for D.
for each $c \in C$ do
    compute P(c) from training set.
end for
for each pair of attributes $A_i$ and $A_j$ do
    for each assignment $a_i$ , $a_j$ and c to $A_i$ , $A_j$ and C do
        compute $P(a_i \mid a_j$ , c ) from training set
    end for
end for
for each pair of attributes $A_i$ and $A_j$ do
    compute $I_p(A_i; A_j \mid C)$ and $W_{ij}$ from training set
end for
for each attribute $A_i$ do
    compute $W_i$ from training set
end for
for each attributes $A_i$ and $j \neq i$ do
    compute $W_{ij}$
end for
```

2.3.3 Deep Belief Network (DBN)

Restricted Boltzmann Machine (RBM) is an energy-based undirected generative model that uses a layer of hidden variables to model a distribution over visible variables [48, 49]. The undirected model for the interactions between the hidden and visible variables is used to ensure that the contribution of the likelihood term to the posterior over the hidden variables is approximately factorial which greatly facilitates inference [50]. Energy-based model means that the probability distribution over the variables of interest is defined through an energy function. It is composed from a set of observable variables $V = \{v_i\}$ and a set of hidden variables $H = \{h_j\}$, i node in the visible layer, j node in the hidden layer. It is restricted in the sense that there are no visible-visible or hidden-hidden connections.

The steps of the RBM learning algorithm can be declared as follows:

1. Due to the conditional independence (no connection) between nodes in the same layer (Property in RBM), the conditional distributions are given in Equations (1) and (2).

$$\begin{cases} P(H|V) = \prod_j p(h_j|v) \\ p(h_j = 1|v) = f(a_i + \sum_i w_{ij}v_i) \\ p(h_j = 0|v) = 1 - p(h_j = 1|v); \end{cases} \tag{2.9}$$

And

$$\begin{cases} P(H|V) = \prod_i p(v_i|h) \\ p(v_i = 1|h) = f(b_j + \sum_j w_{ij}h_j) \\ p(v_i = 0|h) = 1 - p(v_i = 1|h); \end{cases} \tag{2.10}$$

Where f is a sigmoid function (σ) which takes the form $\sigma(z) = 1/1 + e^{-z}$ for binary data vector.

2. The distribution (likelihood) between hidden and visible units is defined as:

$$P(v,h) = \frac{e^{-E(v,h)}}{\Sigma_i e^{-E(v_i,h)}} \tag{2.11}$$

Where $E(x,h)$=$-\bar{h}wv$$-\bar{b}v$$-\bar{c}h$, and $\bar{h}$,$\bar{b}$, $\bar{c}$ are the transposes of matrices h, b and c.

3. The average of the log likelihood with respect to the parameters is given by

$$\begin{aligned} \Delta w_{ij} &= \varepsilon^*(\delta \log p(v)/\delta w_{ij}) \\ &= \varepsilon(< x_i h_j >_{data} - < v_i h_j >_{model}) \end{aligned} \tag{2.12}$$

$$\Delta v_i = \varepsilon(< v_i^2 >_{data} - < v_i^2 >_{model}) \tag{2.13}$$

$$\Delta h_i = \varepsilon(< h_i^2 >_{data} - < h_i^2 >_{model}) \tag{2.14}$$

4. The term $<>_{model}$ takes exponential time to compute exactly so the Contrastive Divergence (CD) approximation to the gradient is used instead [97]. Contrastive divergence is a method that depends on the approximation that is to run the sampler for a single Gibbs iteration, instead until the chain converges. In this case the term $<>_1$ will be used

such that it represents the expectation with respect to the distribution of samples from running the Gibbs sampler initialized at the data for one full step, the new update rule will be.

$$\Delta w_{ij} = \varepsilon(< v_i h_j >_{data} - < v_i h_j > 1) \tag{2.15}$$

$$\Delta v_i = \varepsilon(< v_i^2 >_{data} - < v_i^2 > 1) \tag{2.16}$$

$$\Delta h_i = \varepsilon(< h_i^2 >_{data} - < h_i^2 > 1) \tag{2.17}$$

The Harmonium RBM is an RBM with Gaussian continuous hidden nodes [97]. Where f is normal distribution function which takes the form shown in Equation (10)

$$P(h_j = h|x) = N(c_j + w_j.x, 1) \tag{2.18}$$

Harmonium RBM is used for a discrete output in the last layer of a deep belief network in classification.

The key idea behind training a deep belief network (DBN) by training a sequence of RBMs is that the model parameters θ , learned by an RBM define both $p(v \mid h, \theta)$ and the prior distribution over hidden vectors, $p(h \mid \theta)$, so the probability of generating a visible vector, v, can be written as:

$$p(v) = \Sigma_h p(h \mid \theta).p(v \mid h, \theta) \tag{2.19}$$

After learning θ, $p(v \mid h, \theta)$ is kept while $p(h \mid \theta)$ can be replaced by a better model that is learned by treating the hidden activity vectors $H = h$ as the training data (visible layer) for another RBM. This replacement improves a variation lower bound on the probability of the training data under the composite model. The study in [51] proves the following three rules:

1. Once the number of hidden units in the top level crosses a threshold; the performance essentially flattens at around certain accuracy.

2. The performance tends to decrease as the number of layers increases.

3. The performance increases as we train each RBM for an increasing number of iterations.

In case of not using class labels and back-propagation in the DBN Architecture (unsupervised training) [48], DBN could be used as a feature extraction method for dimensionality reduction. On the other hand, when associating class labels with feature vectors, DBN is used for classification. There are two general types of DBN classifier architectures which are the Back-Propagation DBN (BP-DBN) and the Associate Memory DBN (AM-DBN) [100].

2.3.4 Genetic Algorithm (GA)

Genetic algorithm (GA) is an adaptive search technique initially introduced by Holland [52]. It is computational model designed to simulate the evolutionary processes in the nature [53].

The GA is a stochastic global search method that operates on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximated solution is created by selecting individuals according to their level of fitness in the problem domain and producing offspring from them. This process leads to the evolution of individuals that are better suited to problem domain than the individuals that were created from. Where, the fitness function is used to provide a measure of how individuals have performed in the problem domain.

Individuals are encoded into a finite length strings ”chromosomes”, the most commonly used representation in GAs is the binary alphabet {0, 1}; and other representations can be used is a list of integers.

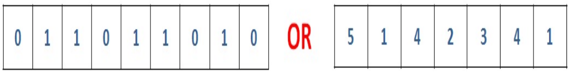


Figure 2.6: Genetic Individuals encoding (chromosome).

The GA begins with a population of random individuals. Then, the population is operated by three main operations; selection, crossover and mutation to create a new population [54].

1. **Selection Operation:**

   A population is created with a group of randomly individuals "chromosome". GA uses fitness function information to assess the performance of the chromosome representation in the problem domain. In the natural world, this would be an individual's ability to survive in its present environment. Thus, the fitness function establishes the basis for the selection of pairs of individuals.

   The selection operator is formulated to selects the better strings in a population and forms a mating pool. It ensures that higher fitness members (better members) have a greater probability of being selected for mating , and that worse members of the population still have a small probability of being selected. This is important to ensure that the search process is global and does not simply converge to the nearest local optimum.

   There exist a number of selection methods in GA literature, but the essential idea in all of them is that the above average strings are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner. The most commonly used methods of selection is Roulette-Wheel Selection [55].

2. **Crossover Operation:**

   Crossover operation is operate at string level, where it recombine two strings to get a better string. Thus, at crossover operation different individuals is created in the successive generations. Where, it randomly select two parents from the mating pool for the crossover operations. Then, it chooses a point randomly in the two selected parents and exchanging the remaining segments of them to create the new individuals [56]. Many crossover operations exist in the GA literature; the most common are One site crossover and two site crossover. Where, One site crossover is more suitable when string length is small while two site crossover is suitable for large strings.

   In one site crossover, a crossover site is selected randomly and all the bits right of the selected site of these two strings are exchanged to form a new pair of strings. The one site crossover operation is shown in fig 2.7

   Two site crossover differ from the one site crossover, where two crossover sites are chosen and the bits between the sites are exchanged. Fig 2.8 shows the two sit crossover operation.

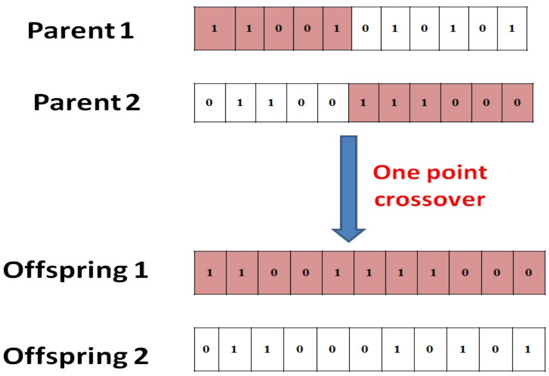


Figure 2.7: One Site Crossover Operation.

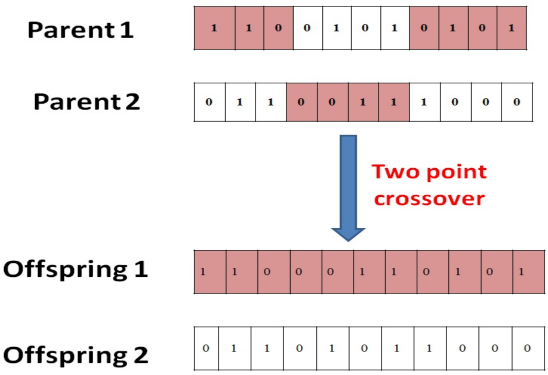


Figure 2.8: Two Site Crossover Operation.

3. **Mutation Operation:**

Mutation operation is operate at the bit level; where it randomly changes one or more bit of a selected individual. This process continues until a suitable solution has been found or a certain number of generations have passed [56].

The most common mutation operations are the bit mutation and swap mutation. The bit mutation is used with the binary representation. Where, the value of the binary bit flips at the location selected to be mutation point. The Binary bit mutation is shown in fig 2.9.

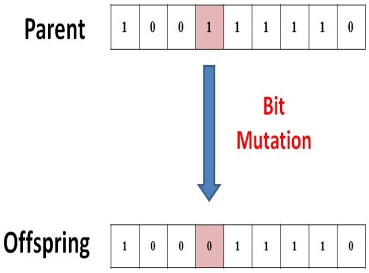


Figure 2.9: Bit Mutation.

While, at the swap mutation two positions are picked up and their values are exchanged; as shown in fig 2.10

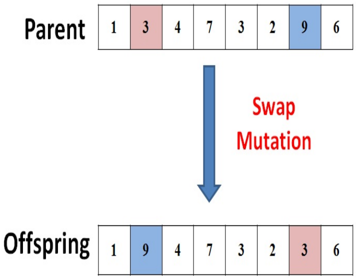


Figure 2.10: Swap Mutation.

Given a well bounded problem GAs can find a global optimum which makes them well suited to feature selection problems. Algorithm 3 shows the structure of a simple Genetic Algorithm (GA).

Algorithm 3 GA algorithm

```
Initialize a population of randomly individuals.
Evaluate population members based on the fitness function.
while termination condition or maximum number of generation Not reach. do
    Select parents from current population.
    Apply crossover and mutation to the selected parents.
    Evaluate offspring.
    Update current population to equal offspring.
end while
Return the best individuals.
```

2.3.5 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [57]. PSO simulates the social behavior of organisms, such as bird flocking. PSO is initialized with a random population (swarm) of individuals (particles). Where, each particle of the swarm represents a candidate solution in the d-dimensional search space. To discover the best solution, each particle changes its searching direction according to:The best previous position (the position with the best fitness value) of its individual memory (pbest), represented by $P_i = (p_{i1}, p_{i2}, ..., p_{id})$; and the global best position gained by the swarm (gbest) $G_i = (g_{i1}, g_{i2}, ..., g_{id})$ [58].

The d-dimensional position for the particle i at iteration t can be represented as:

$$x_i^t = x_{i1}^t, x_{i2}^t, ..., x_{id}^t \tag{2.20}$$

While, the velocity (The rate of the position change) for the particle i at iteration t is given by

$$v_i^t = v_{i1}^t, v_{i2}^t, ..., v_{id}^t \tag{2.21}$$

All of the particles have fitness values, which are evaluated based on a fitness function:

$$Fitness = \alpha.\gamma_R(D) + \beta\frac{|C| + |R|}{|C|} \tag{2.22}$$

Where, $\gamma_R(D)$ is the classification quality of condition attribute set R relative to decision D and $|R|$ is the length of selected feature subset. $|C|$ is the total number of features. While, the parameters α and β are correspond to the importance of classification quality and subset length, $\alpha = [0, 1]$ and $\beta = 1 - \alpha$.

The particle updates its velocity according to:

$$v_{id}^{t+1} = w \times v_{id}^{t} + c_1 \times r_1(p_{id}^{t} - x_{id}^{t}) + c_2 \times r_2(g_{id}^{t} - x_{id}^{t}) \tag{2.23}$$

$$d = 1, 2, ..., D$$

Where, w is the inertia weight and r_1 and r_2 are random numbers distributed in the range [0, 1]. positive constant c_1 and c_2 denotes the cognition learning factor (the private thinking of the particle itself) and the social learning factor (the collaboration among the particles). p_{id}^{t} denotes the best previous position found so far for the i^{th} particle and g_{id}^{t} denotes the global best position thus far [59].

Each particle then moves to a new potential position based on the following equation:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{2.24}$$

$$d = 1, 2, ..., D$$

At every iteration of the PSO algorithm, each particle X_i is updated by the two best values pbest and gbest. Where, pbest denotes the best solution the particle X_i has achieved so far, and gbest denotes the global best position so far. Algorithm 4 shows the main steps of the PSO algorithm-based feature selection.

Algorithm 4 Particle Swarm Optimization Algorithm

```
Input:
m: the swarm size.
$c_1$ , $c_2$ : positive acceleration constants.
w: inertia weight.
MaxGen: maximum generation.
MaxFit: fitness threshold.
Output:
Global best position

Initialize a population of particles with random positions and velocities on D features
  dimensions $pbest_i$=0, Gbest=0, Iter=0.
while Iter < MaxGen or gbest < MaxFit do
  for i = 1 to number of particles m do
    Fitness(i)=Evaluate(i)
```

Algorithm 5 Particle Swarm Optimization Algorithm (cont.)

```
        if fitness(i) > fitness ($pbest_i$) then
            fitness ($pbest_i$)= fitness(i)
            Update $p_{id} = x_{id}$
        end if
        if fitness(i) > Gbest then
            Gbest=Fitness(i)
            Update gbest = i
        end if
        for each dimension d do
            Update the velocity vector.
            Update the particle position.
        end for
    end for
    Iter= Iter+1
end while
Return the Global best position.
```

2.3.6 Fuzzy logic (FL)

In 1965, Lofti Zadeh introduced the fuzzy sets theory, which allows an element to belong to a set with a degree of membership (not only a binary degree) [60][163]. This concept is extended to the Classical logic by Zadeh in 1975 [61] to introduce Fuzzy logic (FL). Classical logic deals with propositions which are either true or false but not both or in between. However, in real world scenario there are cases where propositions can be partially true and partially false. Fuzzy logic handles such real world uncertainty ”vague” by allowing partial truth values.

Fuzzy logic is a methodology for expressing operational laws of a system in linguistic terms instead of mathematical equations. As shown in Figure 2.11 A fuzzy logic system (FLS) consists of four main part [62]:

1. **Fuzzifier:** in Fuzzification process a crisp set of input data are converted to a fuzzy set using fuzzy linguistic variables and membership functions.

   ”Linguistic variables” are the input or output variables of the system whose values are words or sentences from a natural language, instead of numerical values.

"Membership functions" map the non-fuzzy input values to fuzzy linguistic terms and vice versa. Membership functions are used in the fuzzification and defuzzification process.

2. **Fuzzy Rules:** A fuzzy rule is a simple IF-THEN rule with a condition and a consequent.

   IF condition **THEN** consequent

3. **Inference engine:** the Inference process or rule definition portion of fuzzy logic is made based on a set of the fuzzy rules. At this phase, A membership functions have been defined for input and output variables, a control rule base can be developed to relate the output actions of the controller to the observed inputs. Any number of rules can be created to define the actions of the fuzzy controller.

4. **Defuzzifier:** the defuzzification process uses the membership functions to map the resulting fuzzy output to a crisp output.

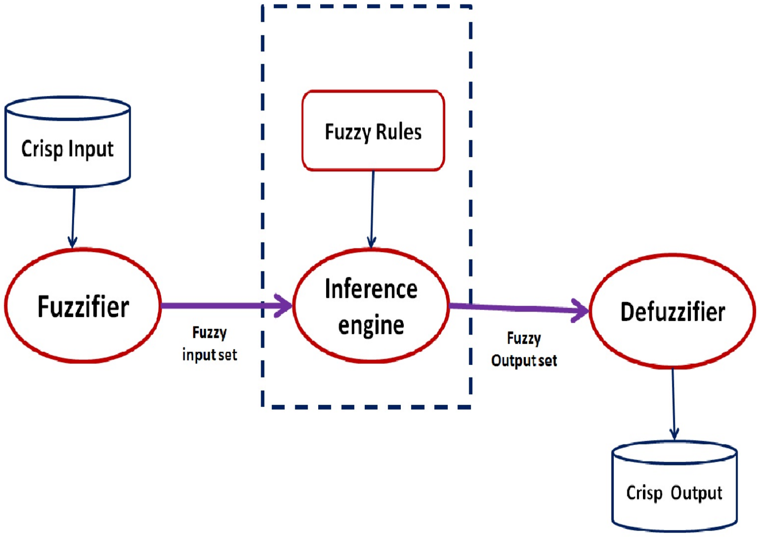


Figure 2.11: fuzzy logic system.

Membership Functions

Membership functions are used in the fuzzification and defuzzification steps of a Fuzzy Logic System. A membership function is used to quantify a linguistic term. The type of the membership function is generally chosen arbitrarily according to the user experience [62]. There are different forms of membership functions, however the most common types of membership functions are triangular, trapezoidal, and Gaussian shapes:

- A triangular membership function is specified by three parameters $\{a, b, c\}$ as follows:

$$triangular(x : a, b, c) = \begin{cases} 0, x \leq a \\ \frac{(x-a)}{(b-a)}, a \leq x \leq b \\ \frac{(c-x)}{(c-b)}, b \leq x \leq c \\ 0, c \leq x \end{cases} \tag{2.25}$$

Where, The parameters $\{a, b, c\}$ with $a < b < c$ determine the x coordinates of the three corners of the underlying triangular membership function.

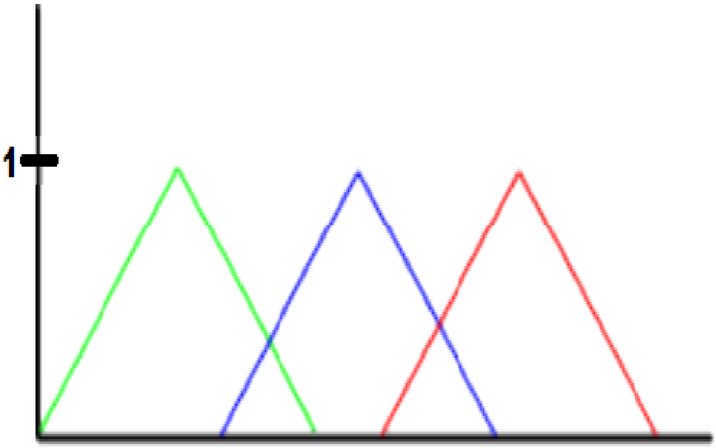


Figure 2.12: Triangular Membership Function.

- A trapezoidal membership function is specified by four parameters $\{a, b, c, d\}$ as follows:

$$trapezoidal(x : a, b, c, d) = \begin{cases} 0, x \leq a \\ \frac{(x-a)}{(b-a)}, a \leq x \leq b \\ 1, b \leq x \leq c \\ \frac{(d-x)}{(d-c)}, c \leq x \leq d \\ 0, d \leq x \end{cases} \tag{2.26}$$

Where, The parameters $\{a, b, c, d\}$ with $a < b <= c < d$ determine the x coordinates of the four corners of the underlying trapezoidal membership function.

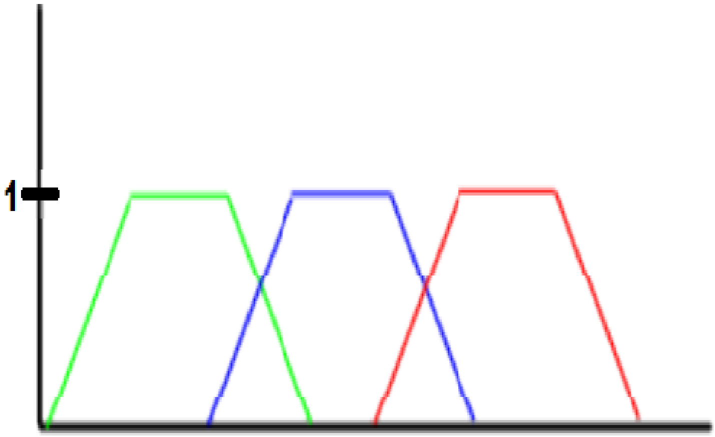


Figure 2.13: Trapezoidal Membership Function.

- A Gaussian membership function is specified by tow parameters $\{c, \sigma\}$ as follows:

$$gaussian(x : c, \sigma) = e^{-\frac{1}{2}(\frac{x-c}{\sigma})^2} \tag{2.27}$$

Where, The parameters c represents the membership function center and σ determines the membership function width.

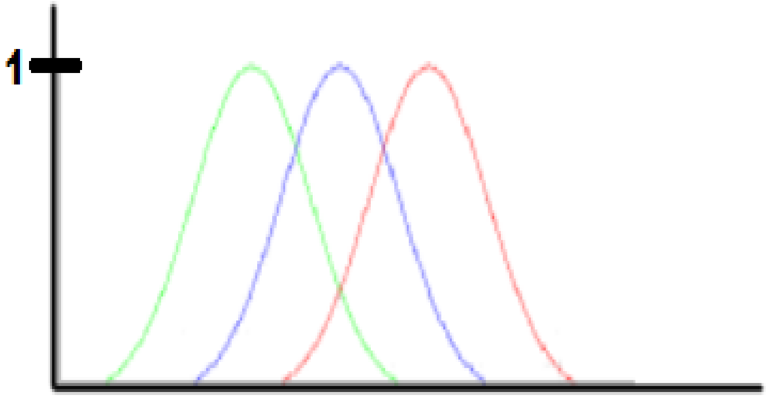


Figure 2.14: Gaussian Membership Function.

The Algorithm of fuzzy logic is given in Algorithm 6:

Algorithm 6 Fuzzy Logic Algorithm

```
Define the linguistic variables
Construct the membership functions
Construct the Fuzzy rule base
Fuzzification: Convert crisp input data to fuzzy values using the membership functions
Inference: Evaluate the rules in the rule base
Combine the results of each rule
Defuzzification: Convert the output data to non-fuzzy values
```

2.4 Data Preprocessing

2.4.1 Data Reduction

Network data sets may contain irrelevant or redundant features. Extraneous features can make it harder to detect suspicious behavior patterns, causing the curse of dimensionality problem [63].Therefore, data reduction must be performed to high dimensional data set; to improve the classification performance and reduces the computational cost.

Data dimensionality reduction can be achieved in two different ways: feature extraction and feature selection. Feature extraction methods create a new set of features by linear or nonlinear combination of the original features. While, feature selection methods generate a new set of features by selecting only a subset of the original features [64].

Feature selection aims to choose an optimal subset of features that are necessary to increase the predictive accuracy and reduce the complexity of learned results[65, 66]. Different feature selection methods are proposed to enhance the performance of IDS [67].

Based on the evaluation criteria feature selection methods fall into two categories: filter approach [65, 68] and wrapper [69, 70] approach.

- **Filter approaches** evaluate and select the new set of features depending on the general characteristics of the data without involving any machine algorithm. The features are ranked based on certain statistical criteria, where features with highest ranking values are selected. Frequently used filter methods include chi-square test [71], information gain [72], and Pearson correlation coefficients [73].

- **Wrapper approaches** use a predetermined machine algorithm and use the classification performance as the evaluation criterion to select the new features set. Machine learning algorithms such as ID3 [74] and Bayesian networks [75] are commonly used as induction algorithm for wrapper approaches.

  The advantages of filter based approaches are the low computational cost and the independent of the learning algorithm. Thus, they can easily scale up to high-dimensional datasets [76].

2.4.2 Feature Selection Methods

2.4.2.1 Principle Components Analysis (PCA)

It is well known that principal component analysis (PCA) is an essential technique in data compression and feature extraction [77], and it has been also applied to the field of ID [12, 78, 79]. It is well known that PCA has been widely used in data compression and feature selection. Feature selection refers to a process whereby a data space is transformed into a feature space, which has a reduced dimension. Some basic knowledge of PCA is briefly described in the next.

Assume that $\{x_t\}$ where $t = 1, 2 \ldots, N$ are stochastic n-dimensional input data records with mean (μ). It is defined by the following Equation:

$$\mu = \frac{1}{N} \sum_{t=1}^{N} x_t \tag{2.28}$$

The covariance matrix of x_t is defined by

$$C = \frac{1}{N} \sum_{t=1}^{N} (x_t - \mu).(x_t - \mu)^T \tag{2.29}$$

PCA solves the following eigenvalue problem of covariance matrix C:

$$Cv_i = \lambda_i v_i \tag{2.30}$$

where λ_i $(i = 1, 2, ..., n)$ are the eigenvalues and $v_i(i = 1, 2, ..., n)$ are the corresponding eigenvectors.

To represent data records with low dimensional vectors, we only need to compute the m eigenvectors (called principal directions) corresponding to those m largest eigenvalues ($m < n$). It is well known that the variance of the projections of the input data onto the principal direction is greater than that of any other directions.

Let

$$\phi = [v_1, v_2, ..., v_m], \Lambda = diag[\lambda_1, \lambda_2, ..., \lambda_m] \tag{2.31}$$

Then

$$C\Phi = \Phi\Lambda \tag{2.32}$$

The parameter v denote to the approximation precision of the m largest eigenvectors so that the following relation holds.

$$\frac{\Sigma_{i=1}^{m}\lambda_i}{\sum_{i=1}^{n}\lambda_i} \geq v \tag{2.33}$$

Based on (7) and (8) the number of eigenvectors can be selected and given a precision parameter v, the low-dimensional feature vector of a new input data x is determined by

$$x_f = \Phi^T x \tag{2.34}$$

2.4.2.2 Information Gain (IG)

The information gain (IG) [80, 81] of a given attribute X with respect to the class attribute Y is the reduction in uncertainty about the value of Y, after observing values of X. It is given by

$$IG = Y \mid X \tag{2.35}$$

When Y and X are discrete variables that take values in $y_1...y_k$ and $x_1...x_l$ then the uncertainty about the value of Y is measured by its entropy

$$H(Y) = -\sum_{i=1}^{k} P(y_i)log_2(P(y_i)) \tag{2.36}$$

where $P(y_i)$ is the prior probabilities for all values of Y.

The uncertainty about the value of Y after observing values of X is given by the conditional entropy of Y given X

$$H(Y \mid X) - \sum_{j=l}^{n} P(x_j) \sum_{i=1}^{k} P(y_i \mid x_j)log_2(P(y_i \mid x_j)) \tag{2.37}$$

where $P(y_i \mid x_j)$ is the posterior probabilities of Y given the values of X.

Thus, the information gain is given by:

$$IG(Y \mid X) = H(Y) - H(Y \mid X) \tag{2.38}$$

Following this measure, an attribute X is regarded more correlated to class Y than attribute Z, if $IG(Y \mid X) > IG(Y \mid Z)$.

we can rank the correlations of each attribute to the class and select key attributes based on the calculated information gain [82].

2.4.3 Data Discretization

Discretization is a process of converting the continuous space of features into a nominal space [83]. The goal of the discretization process is to find a set of cut points, which split the range into a small number of intervals. Each cut-point is a real value within the range of the continuous values, which divides the range into two intervals one greater than the cut-point and other less than or equal to the cut-point value [84]. Discretization is usually performed as a pre-processing phase to the learning algorithm.

Discretization methods can be classified into five categories [85]:

1. Supervised vs. Unsupervised
2. Static vs. Dynamic
3. Global vs. Local
4. Top-down (splitting) vs. Bottom-up (merging)
5. Direct vs. Incremental

Supervised methods use the class labels during the discretization process. In contrast, Unsupervised methods do not use information about the class labels and generate discretization schemes based only on distribution of the values of the continuous attributes. Researches show that supervised methods are better than unsupervised methods [86]. Dynamic and static methods depends on whether the method considers the interdependence among the features into account or not [87]. Global methods use the entire value space of a numeric attribute for the discretization. While, Local methods use a subset of instances when deriving the discretization. Top-down(splitting) discretization methods start with one interval of all values of feature and split it into smaller intervals at the subsequent iterations. While, the bottom-up (merging) methods start with the maximal number of sub-intervals and merge these sub intervals until achieving a certain stopping criterion or optimal number of intervals [88]. Direct methods divide the range into equal-width of intervals, it requires the user to determine the number of intervals. Incremental methods begin with a simple discretization and go through an improvement process until reaching a stopping criterion to terminate the discretization process [89].

Fayyad et al. [90] proposed the Information Entropy Maximization (IEM) discretization method. It is a supervised, local, splitting and incremental discretization method. IEM algorithm criterions are based on information entropy, where the cut points should be set between points with different class labels.

Let T partition set S into subsets S_1 and S_2, for k classes $C_1, ..., C_k$ the class entropy of a subset S is given by

$$Ent(S) = -\sum_{i=1}^{k} P(C_i, S) log(P(C_i, S)) \tag{2.39}$$

where $P(C_i, S)$ is the proportion of examples in S that have class C_i.

For an attribute A, the class information entropy of the partition induced by partition T is defined as

$$E(A, T; S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2) \tag{2.40}$$

2.5 Chapter Conclusion

It is stated that Computational Intelligence methods provide the framework of designing anomaly NIDS to secure networks channels. the CI methods can be hybrid with different data preprocessing techniques as feature selection and discritization for building fast accurate NIDS. A brief introduction of CI, CI methods, feature selection methods and data discritization are presented in this chapter.

CHAPTER 3

The Proposed hybrid NID models

Chapter 3

The Proposed Hybrid NID Models

3.1 Introduction

Intrusion detection systems (IDSs) is an essential key for network defense. However, many issues needs to be consider when building an IDS, such as data preprocessing, classification accuracy (detection precision) and detection speed. In this chapter, we propose four anomaly hybrid NIDS as shown in Fig 3.1. The first three proposed models (PCA-SVM, DBN-SVM and GA-HNB) combine the individual base classifiers with feature selection algorithm to maximize the detection accuracy and minimize the computational complexity. While, the fourth proposed NIDS (IEM-GA-Classifier) combines the classifiers with two data preprocessing methods (discretization and FS).

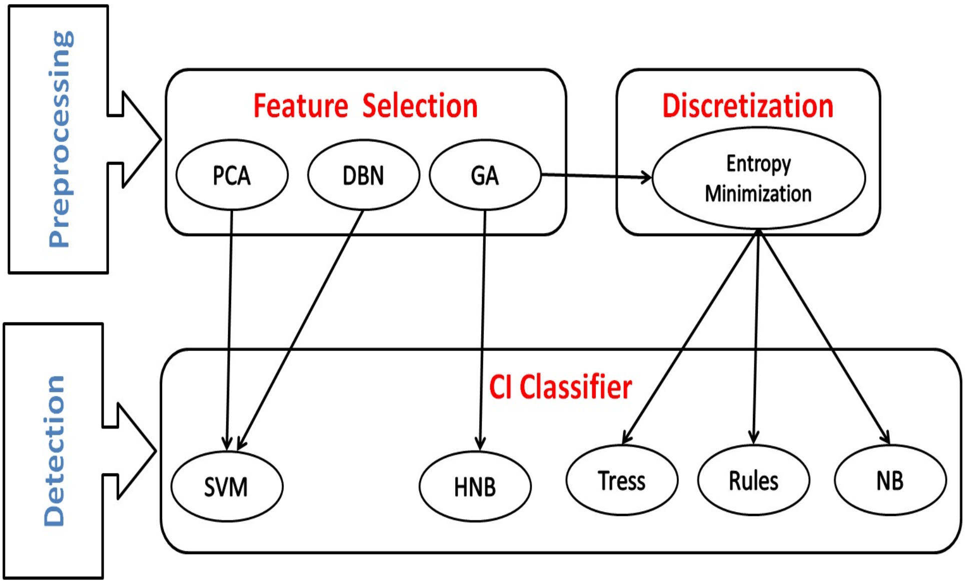


Figure 3.1: The Four Proposed hybrid NID models

3.2 Intrusion Detection System

Intrusion detection system (IDS) dynamically monitors the events taking place in a system, and decides whether these events are symptomatic of an attack (intrusion) or constitute a legitimate use of the system [91].

3.2.1 Taxonomy of Intrusion Detection System

There are several ways to categorize an IDS depending on, the location of the IDS in the system and the detection methodology used to generate alerts, figure 3.2.

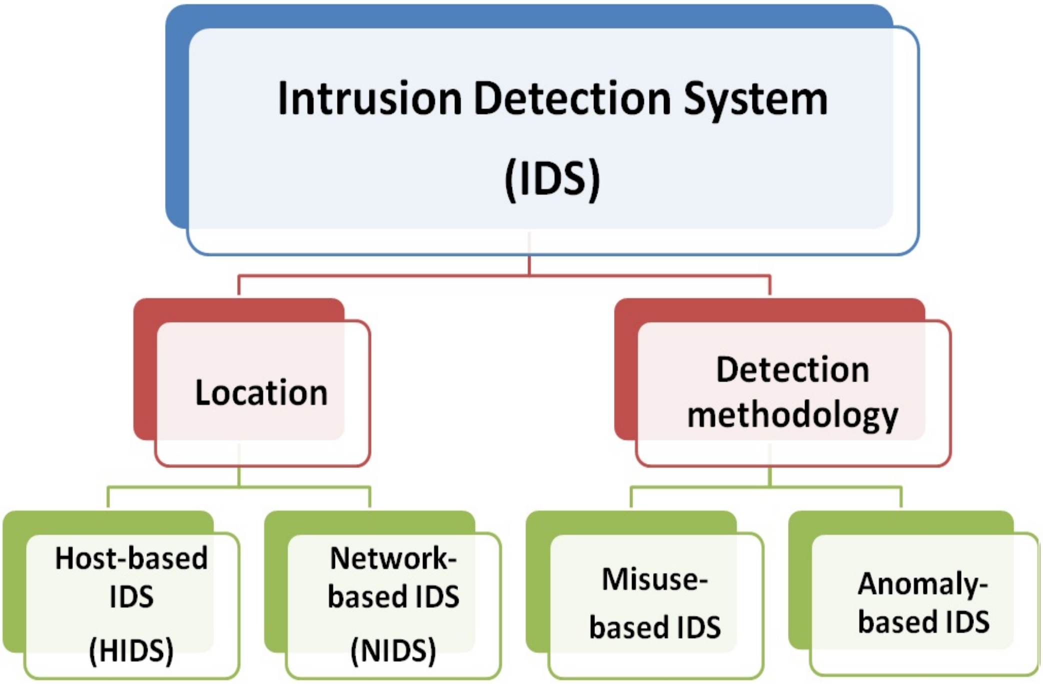


Figure 3.2: Intrusion Detection System Taxonomy

3.2.1.1 Host-based vs. Network-based IDS

Host-based Intrusion Detection System (HIDS) were the first type of IDS to appear [92]. HIDS are installed on the host and monitors the operating system information (eg. system call sequences and application logs) [91]. HIDSs have the advantage of being able to detect attacks from the inside,by checking traffic before being sent or just received. However, The main problem of HIDSs is that they can only monitor the single host they are running on, and have to be specifically set up for each host. Thus, scalability is the main issue for HIDSs [93, 94].

Network-based Intrusion Detection System (NIDS) identifies intrusions on external interfaces for network traffic among multiple hosts. NIDS gains access to network traffic by placing detection sensors (sniffers) at hubs or network switches to monitor packets traveling among various communication mediums. The main advantage is that a single system can be used to monitor the whole network (or part of it), without the need of installing a dedicated software on each host. However, NIDS main disadvantages is that they can have difficulties when processing large amount of network packets [95].

3.2.1.2 Misuse-based IDS vs. Anomaly-based IDS

Depending on the analysis techniques IDSs can be divided into two techniques: misuse detection and anomaly detection [96, 97].

Misuse intrusion detection system(signature-based detection) is the most popular commercial type of IDSs [98]. A misuse-based detection system contains a database which includes a number of signatures about known attacks. The audit data collected by the IDS is compared with the well-defined patterns of the database and, if a match is found, an alert is generated [99, 100]. The main drawbacks of misuse-based detection is that it is not able to alert the system administrator in case of new attacks [94].

Anomaly intrusion detection is a behavior-based detection method . It based on the idea of building a normal traffic profiles [101]. It identifies malicious traffic based on the deviations from the normal profiles, where the normal patterns are constructed from the statistical measures of the system features [102, 103]. One of the key advantages of anomaly based IDS over the misuse detection; is that it can detecte new threats, or different versions of known threats [104, 105].

3.3 Network intrusion Dataset

Under the sponsorship of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), Massachusetts Institute of Technology (MIT) Lincoln Labs [106] in 1998 and 1999 collected and distributed the 1998 DARPA dataset and 1999 DARPA dataset for Intrusion Detection Evaluation (IDEVAL) [107, 108, 109, 110]. The 1998 DARPA dataset includes 7 weeks of training data with labelled test data and 2 weeks of unlabelled test data , It contains over 300 instances of 38 attacks [107]. The 1999 DARPA dataset presents over 5 million connections over 5 weeks: 2 were attack-free and 3 weeks included attacks.

KDD'99 dataset is a subset of the DARPA benchmark dataset prepared by Sal Stofo and Wenke Lee [111]. The KDD'99 train dataset is about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic, processed into about five million connection Record each with about 100 bytes. The two weeks of test data have around two million connection records. Each KDD'99 training connection record contains 41 features. There are 38 numeric features and 3 symbolic features, falling into the following four categories:

1. **Basic features**: 9 basic features describe each individual TCP connection.

2. **Content features**: 13 domain knowledge related features which indicate suspicious behavior in the network traffic. This includes features such as the number of failed login attempts.

3. **Time-based traffic features**: 9 features used to summarize the connections over 2 second temporal window. Such as the number of connections that had the same destination host or the same service.

4. **Host-based traffic features**: 10 features constructed using a window of 100 connections to the same host instead of a time window. It is designed to assess attacks, which span intervals longer than 2 seconds.

Each record is labeled as either normal or an attack, with exactly one specific attack type. The training set contains a total of 22 training attack types, with an additional 17 types in the testing set only.

The attacks fall into four categories:

1. **Denial of service (DoS)**: attacker tries to prevent legitimate users from using a service. e.g Neptune, Smurf, Pod and Teardrop.

2. **Remote-to-Local (R2L)**: unauthorized access to local from a remote machine e.g Guess-password, Ftp-write, Imap and Phf.

3. **User-to-Root (U2R)**: unauthorized access to root privileges e.g Buffer-overflow, Load-module, Perl and Spy.

4. **Probing** eg. Port-sweep, IP-sweep, Nmap and Satan.

Table 3.1 gives a description of KDD'99 Intrusion Dataset Features and their data types [112].

Table 3.1: KDD'99 dataset Features and their data types

Feature	Description	Type
1. duration	Duration of the connection.	Cont.
2. protocol type	Connection protocol (e.g. tcp, udp)	Disc.
3. service	Destination service (e.g. telnet, ftp)	Disc.
4. flag	Status flag of the connection	Disc.
5. source bytes	Bytes sent from source to destination	Cont.
6. destination bytes	Bytes sent from destination to source	Cont.
7. land	1 if connection is from/to the same host/port; 0 otherwise	Disc.
8. wrong fragment	number of wrong fragments	Cont.
9. urgent	number of urgent packets	Cont.
10. hot	number of "hot" indicators	Cont.
11. failed logins	number of failed logins	Cont.
12. logged in	1 if successfully logged in; 0 otherwise	Disc.
13. # compromised	number of "compromised" conditions	Cont.
14. root shell	1 if root shell is obtained; 0 otherwise	Cont.
15. su attempted	1 if "su root" command attempted; 0 otherwise	Cont.
16. # root	number of "root" accesses	Cont.
17. # file creations	number of file creation operations	Cont.
18. # shells	number of shell prompts	Cont.
19. # access files	number of operations on access control files	Cont.
20. # outbound cmds	number of outbound commands in an ftp session	Cont.
21. is hot login	1 if the login belongs to the "hot" list; 0 otherwise	Disc.
22. is guest login	1 if the login is a "guest" login; 0 otherwise	Disc.
23. Count	number of connections to the same host as the current connection in the past two seconds	Cont.
24. srv count	number of connections to the same service as the current connection in the past two seconds	Cont.
25. serror rate	% of connections that have "SYN" errors	Cont.
26. srv serror rate	% of connections that have "SYN" errors	Cont.
27. rerror rate	% of connections that have "REJ" errors	Cont.
28. srv rerror rate	% of connections that have "REJ" errors	Cont.

29. same srv rate	% of connections to the same service	Cont.
30. diff srv rate	% of connections to different services	Cont.
31. srv diff host rate	% of connections to different hosts	Cont.
32. dst host count	count of connections having the same destination host	Cont.
33. dst host srv count	count of connections having the same destination host and using the same service	Cont.
34. dst host same srv rate	% of connections having the same destination host and using the same service	Cont.
35. dst host diff srv rate	% of different services on the current host	Cont.
36. dst host same src port rate	% of connections to the current host having the same src port	Cont.
37. dst host srv diff host rate	% of connections to the same service coming from different hosts	Cont.
38. dst host serror rate	% of connections to the current host that have an S0 error	Cont.
39. dst host srv serror rate	% of connections to the current host and specified service that have an S0 error	Cont.
40. dst host rerror rate	% of connections to the current host that have an RST error	Cont.
41. dst host srv rerror rate	% of connections to the current host and specified service that have an RST error	Cont.

Statistical analysis on KDD'99 dataset found important issues which highly affects the performance of evaluated systems and results in a very poor evaluation of anomaly detection approaches.

Leung and Leckie [113] reported two problems in the KDD'99 dataset.

1. KDD'99 dataset contains huge number of redundant records.10% portions of the full dataset contained only two types of DoS attacks (Smurf and Neptune).These two types constitute over 71% of the testing dataset which completely affects the evaluation.

2. since these attacks consume large volumes of traffic, they are easily detectable by other means and there is no need of using anomaly detection systems to find these attacks.

To solve these issues, NSL-KDD a new dataset is suggested [114]. NSL-KDD consists of selected records of the complete KDD'99 dataset,where the repeated records in the entire KDD'99 train and test set are removed.KDD'99 dataset contains 4898431 records in train set and 311027 records in test set.Table 3.2gives statistics of the reduction of repeated records in the KDD train and test sets.The KDD'99 train set is reduced by 78.05% and the test set is reduced by 75.15%.

Table 3.2: KDD'99 dataset reduction statistics

		Train set		Test set
	Repeated records	Reduction rate	Repeated records	Reduction rate
Intrusion	3663472	93.32%	221058	88.26%
Normal	159967	16.44%	12680	20.92%

NSL-KDD dataset has the following advantages over the original KDD'99 dataset [114]:

1. The train set does not include redundant records; hence the classifiers will not be biased towards more frequent records.
2. The proposed test sets have no duplicate records; therefore, the performances of the learners are not biased by the methods which have better detection rates on the frequent records.
3. The number of records in the train and test sets is reasonable, which

makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

3.4 Hybrid PCA-SVM ID model

The proposed Hybrid PCA-SVM ID model is composed of the following three phases.

1. **Preprocessing:** mapping symbolic valued attributes to numeric and scaling data and attack names.
2. **Feature selection:** select the optimum feature subset.
3. **Intrusion detection:** classify the intrusion type into its classes.

Figure 3.3 shows the overall description of the proposed PCA-SVM intrusion detection model.

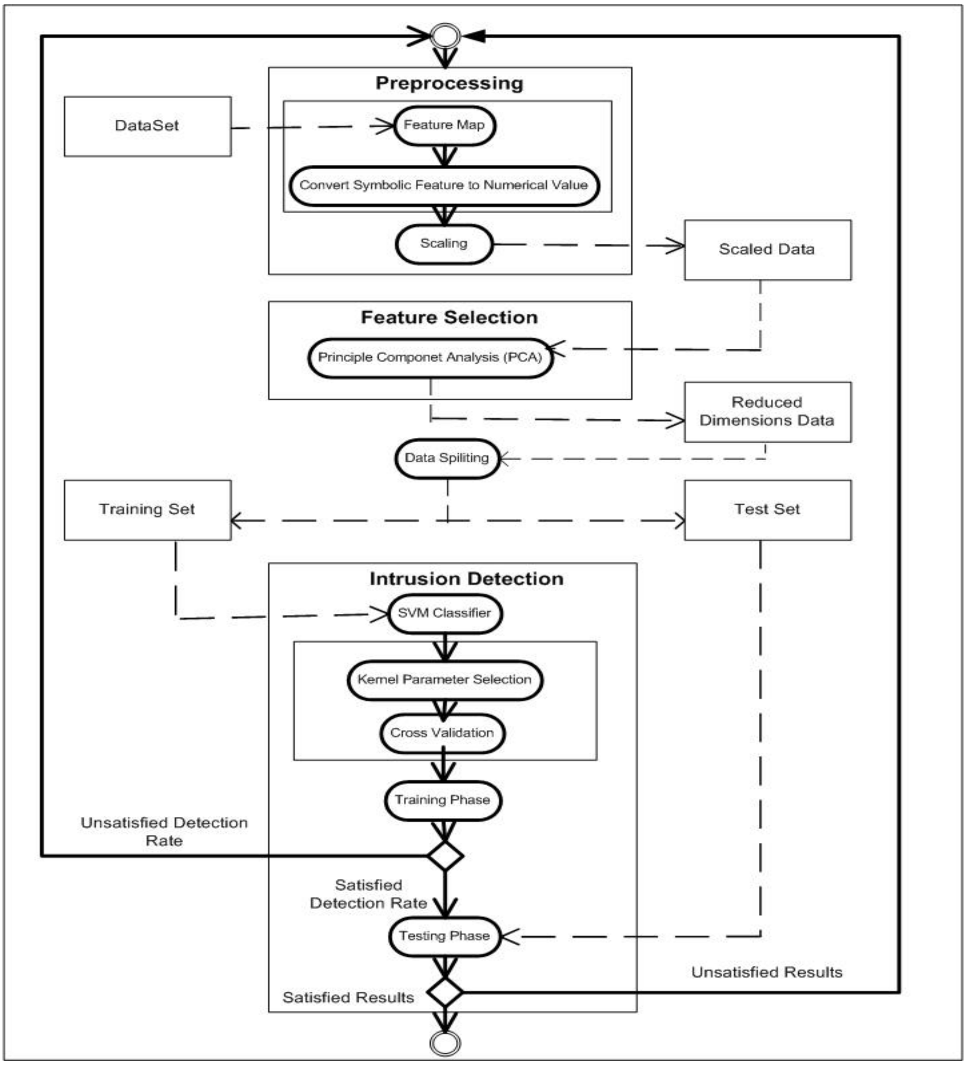


Figure 3.3: The Hybrid PCA-SVM ID model

3.4.1 Preprocessing Phase

SVM classification system are not able to process NSL-KDD dataset in its current format. Hence preprocessing was required before SVM classification system could be built. Preprocessing contains the following processes:

- Mapping symbolic features to numeric value.
- Implementing scaling since the data have significantly varying resolution and ranges. The attribute data are scaled to fall within the range [-1, 1].
- Attack names were mapped to one of the five classes, 0 for $Normal$, 1 for DoS (Denial of Service), 2 for $U2R$ (user-to-root: unauthorized access to root privileges) , 3 for $R2L$ (remote-to-local: unauthorized access to local from a remote machine), and 4 for *Probe* (probing:information gathering attacks) .

3.4.2 Feature Selection Phase

PCA reduces the amount of dimensions required to classify new data and produces a set of principal components, which are orthonormal eigenvalue/ eigenvector pairs [115]. It reduces the dimensionality of data by restricting attention to those directions in the feature space in which the variance is greatest. The proportion of the total variance accounted for a feature is proportional to its eigenvalue [116]. We apply PCA on NSL-KDD dataset to reduce the dimensions (D) of the data where $D = 41$. We use the Critical Eigenvalue test and screeplot test [116] to determine the k features that are required for classification. The Critical Eigenvalue test recommends selecting all principal components whose eigenvalues exceed the threshold $\frac{D^{0.6}}{15}$, where D is the dimension of the original dataset. The remaining d-k features are assumed to contain noise or redundancy. For NSL-KDD dataset we select 23 features which reduce the input data dimension by 56%.

3.4.3 Intrusion Dtection Phase:Parameter Selection

Support vector machine (SVM) The radial basis function kernel (RBF) is used within this study, mainly for some reasons [117]:

- RBF makes it possible to map the non-linear boundaries of the input space into a higher dimensional feature space unlike the linear kernel.
- The RBF kernel has less numerical difficulties because the kernel values lie between zero and one, while the polynomial kernel values may go to infinity or zero while the degree is large.
- When looking at the number of hyper parameters, the polynomial kernel has more hyper parameters than the RBF kernel. On the basis of these arguments, the RBF kernel is used as the default kernel function.

In this study, a K-fold cross-validation is used on the dataset to search the best values of RBF kernel width parameter γ and constant C. The search is realized by evaluating exponential sequences of γ and C.

3.5 Hybrid DBN-SVM ID Model

Figure 3.4, shows the Hybrid DBN-SVM model. DBN in this model is used as a feature reduction method such that the reduced-dimensional output from the DBM is passed to the SVM for classification.

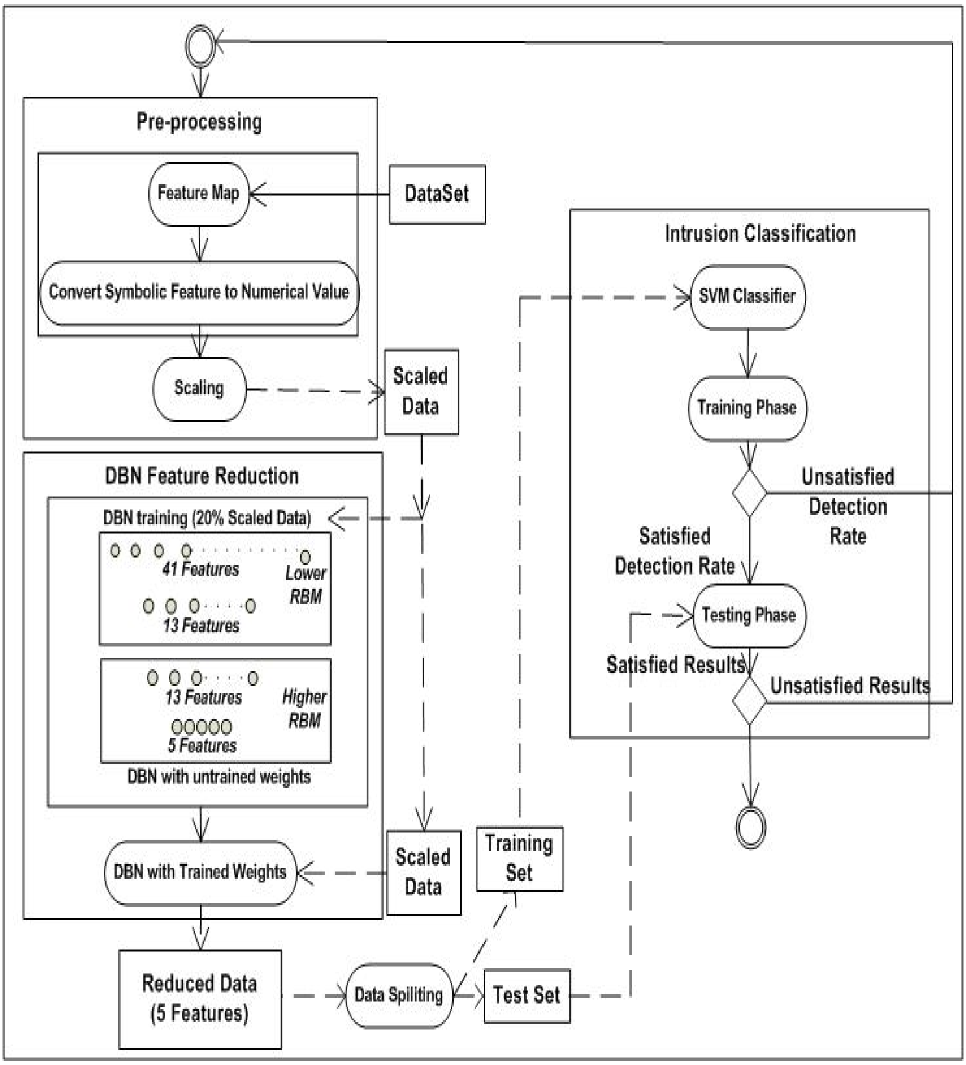


Figure 3.4: DBN-SVM Hybride Model

3.5.1 Preprocessing Phase

Preprocessing phase of NSL-KDD dataset contains the following processes:

- Mapping symbolic features to numeric value.
- Scaling since the data have significantly varying resolution and ranges. The attribute data are scaled to fall within the range [0, 1].
- Attack names were assigned to one of the five classes, 0 for Normal, 1 for DoS (Denial of Service), 2 for U2R (User-To-Root), 3 for R2L remote-to-local, and 4 for Probe.

3.5.2 DBN Feature Reduction Phase

DBN has been used as a data reduction method. The algorithm of the hybrid model works in two main steps as follows:

1. Using DBN as dimensionality reduction method with back-propagation to enhance the reduced data output. The DBN Network has the BP-DBN structure that is constructed of 2 RBM layers, the first RBM layer efficiently reduces the data(e.g. from 41 to 13 feature and the second from 13 features to 5 output features based on NSL-KDD data).
2. Pass the output from the DBN of 5 features to the SVM to be classified.

3.5.3 Intrusion Detection Phase

Support vector machine (SVM) is a classification technique based on Statistical Learning Theory (SLT). It is based on the idea of a hyper plane classifier, where it first maps the input vector into a higher dimensional feature space and then obtains the optimal separating hyper-plane. The goal of SVM is to find a decision boundary (i.e. the separating hyper-plane) so that the margin of separation between the classes is maximized [118].

3.6 Hybrid GA-HNB ID Model

The proposed hybrid anomaly intrusion detection approach is using the advantages of hidden naïve bayes (HNB) in conjunction with genetic algorithm-based feature selection to detect and classify the network intrusions into five outcomes: normal and four anomaly intrusion types. It is comprised of the following three fundamental building phases:

1. **Preprocessing phase**: In the first phase of the investigation, a preprocessing algorithms were used to map symbolic features to numeric value and attack names were mapped to one of the five classes. It is adopted and used to improve the quality of the data and to make the feature selection phase more reliable.

2. **Feature reduction-based genetic algorithm phase**: In the second phase, genetic algorithm has been used as feature selection to reduce

the dimensionality of the dataset.

3. **Detection and classification using hidden naïve bays phase**: The last phase is the intrusion detection and classification of a new intrusion into five outcome. These three phases are described in detail in the following section along with the steps involved and the characteristics feature for each phase.

Figure 3.5 Depicts the overall architecture of the introduced approach.

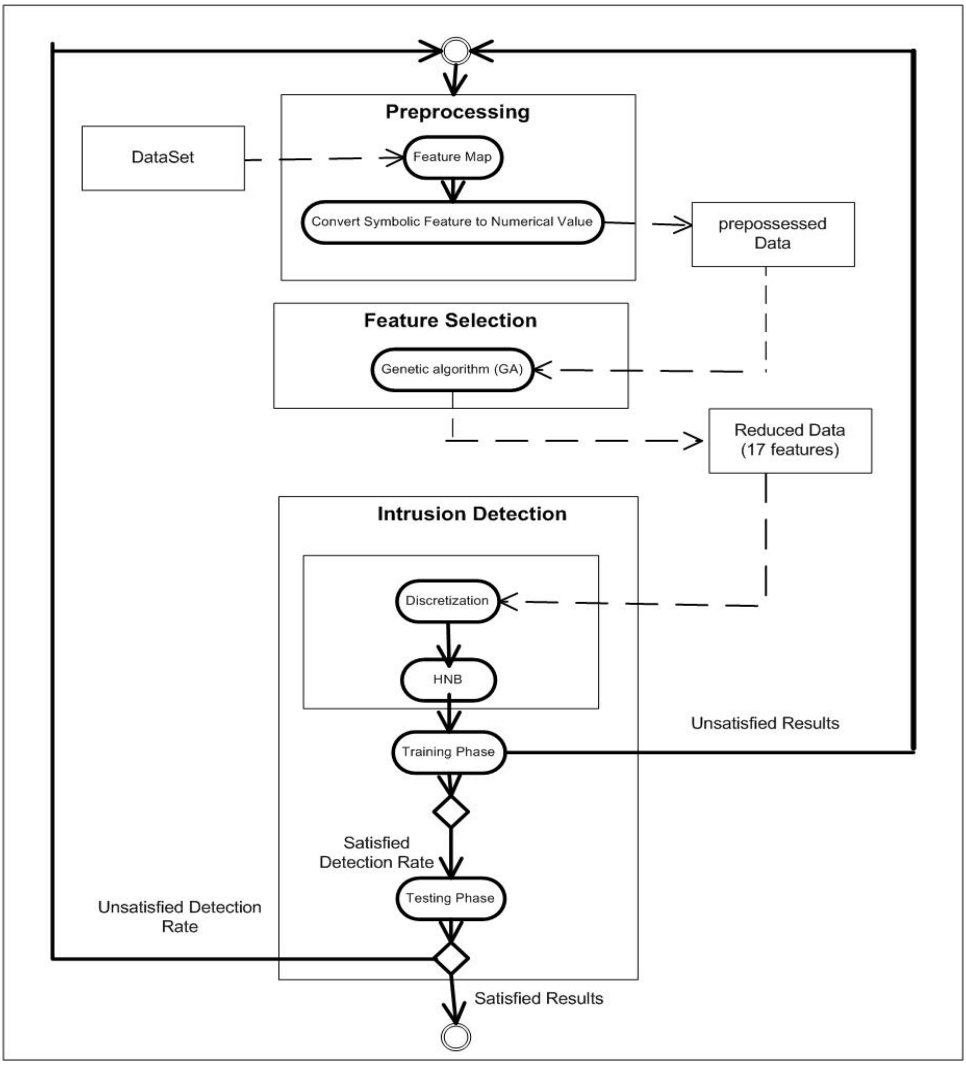


Figure 3.5: The overall architecture of the anomaly hybrid GA-HNB ID approach

3.6.1 Preprocessing Phase

The following two pre-processing stages has been done on NSL-KDD dataset:

1. Mapping symbolic features to numeric value.
2. Attack names were mapped to one of the five classes, 0 for $Normal$, 1 for DoS (Denial of Service), 2 for $U2R$ (user-to-root: unauthorized access to root privileges), 3 for $R2L$ (remote-to-local: unauthorized access to local from a remote machine), and 4 for $Probe$ (probing:information gathering attacks.

3.6.2 GA Feature Selection Phase

Feature selection is a process that reduces the feature space by eliminating unnecessary features to classification. There are two main feature selection approaches, filter methods and Wrapper methods [67]. Filter method does not rely on any particular learning algorithm. It assesses each feature independently, and then a subset of features from the feature space is selected according to the ranking between the features. Wrapper method relies on some learning algorithm to estimate the features subset. Using cross-validation it calculates the accuracy of the learning algorithm for each feature from the feature space to predict the benefits of adding or removing this feature from the feature subset.In this paper, genetic algorithm has been used as feature selection method to reduce the dimensionality of the dataset. GA efficiently reduces the NSL-KDD dataset from 41 features to 17 features, which reduces 58.5% of the feature space.

3.6.3 Intrusion Detection and Classification Phase:HNB Approach

The 17 features output from the GA where discritized by the Entropy Minimization Discretization method [119]. Then, the reduced discritized dataset is passed to the HNB classifier to be classified. The parameters of the HNB classifier are estimated from the training data. The GA-HNB model is described in algorithm 7.

Algorithm 7 hybrid GA-HNB intrusion detection approach

```
Use GA to reduce the training set.
Discritize the reduced training set by Entropy Minimization Discretization method.
Use the output of the GA layer to build HNB classifier.
for each $c \in C$ do
    compute P(c) from training set.
end for
for each pair of attributes $A_i$ and $A_j$ do
    for each assignment $a_i$ , $a_j$ and c to $A_i$ , $A_j$ and C do
        compute $P(a_i \mid a_j$ , c ) from training set
    end for
end for
for each pair of attributes $A_i$ and $A_j$ do
    compute $I_p(A_i; A_j \mid C)$ and $W_{ij}$ from training set
end for
Run the testing dataset through the network (GA-Discritized-trained HNB) to assign
a class label for each instance
if Assigned class label is equal to actual class label then
    object is classifier correctly
end if
Calculate the classification accuracy.
```

3.7 Hybrid Real-Time Discretize ID Model

The framework for the proposed anomaly intrusion detection approach is shown in Fig 3.6. It is comprised of the following three fundamental building phases:

1. Data set Pre-processing by mapping and IEM discretization.

2. Data reduction by GA feature selection.

3. Intrusion detection and classification of a new intrusion into five outcome.

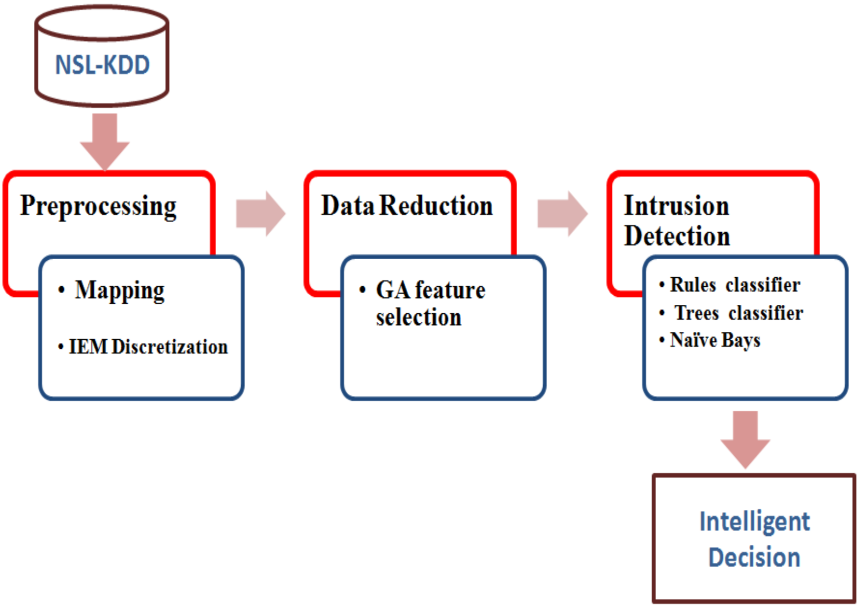


Figure 3.6: Real-time Discretize Network Intrusion Detection Framework

3.7.1 Preprocessing Phase

The following two preprocessing stages has been done on NSL-KDD dataset:

1. **Mapping**:

- symbolic features to numeric value.
- Attack names to one of the five classes, 0 for *Normal*, 1 for DoS (Denial of Service), 2 for *U2R* (user-to-root: unauthorized access to root privileges), 3 for *R2L* (remote-to-local: unauthorized access to local from a remote machine), and 4 for *Probe* (probing:information gathering attacks.

2. **Discretization**: Features where discritized by Information Entropy Maximization (IEM) discretization method.

3.7.2 GA Feature Selection Phase

GA is applied as a feature selection method to reduce the dimensionality of the dataset. GA efficiently reduces the NSL-KDD dataset from 41 features to 14 features, which reduces 65% of the feature space. Algorithm 8 gives the main steps of the genetic algorithm-based feature selection.

3.7.3 Intrusion Detection Phase

we evaluate the performance of the proposed high speed network intrusion detection framework on different set of classifier. The set of classifier includes:

- Rules based classifiers (Ridor, Decision table).
- Trees classifiers (REPTree, C 4.5, Random Forest).
- Naïve bays classifier.

Algorithm 8 Genetic algorithm-based feature selection

```
Initialize a population of randomly individual $M(0)$ of 41 NSL-KDD features.
for Reaching fitness threshold or maximum number of generation do
    Evaluate the fitness $f(m)$ of each individual $m$ in the current population $M(t)$
    select the best-fit individuals using selection probabilities $P(m)$ for each individual $m$ in $M(t)$
    Generate new individuals $M(t+1)$ through crossover and mutation operations to produce offspring.
    Replace least-fit individual with new ones.
end for
Return the best n features of NSL-KDD dataset.
```

3.8 Chapter Conclusion

Several CI techniques in the literatures have been proposed for the design of IDS. In particular, these techniques are developed to classify whether the incoming network trances are normal or intruder. Detection precision and detection speed is a vital key indicator to evaluate an intrusion detection system. However, when dealing with data containing large number of features the classification accuracy and speed are effected. Therefore, feature selection is required to deal with a large feature space. In this chapter, four proposed hybrid anomaly NIDS are proposed that are aiming to solve such problem. First, three hybrid proposed models (PCA-SVM, DBN-SVM and GA-HNB) uses feature selection algorithm as data preprocessing combine with the individual base classifiers; to reduce the computational complexity and remove information redundancy which increase the detection accuracy of the classifiers. Second, the fourth proposed hybrid anomaly NIDS (IEM-GA-Classifier) combines two data preprocessing methods; IEM discretization and GA feature selection; with the base classifiers.

CHAPTER 4

ExperimentalWork & results Evaluation on the proposed Hybrid NIDS models

Chapter 4

Experimental Results and Evaluation on The Proposed Hybrid NIDS Models

4.1 Introduction

This chapter investigate the detection performance of the four proposed Hybrid anomaly NIDS models to classify the network intrusion into five outcomes: normal, and four anomaly types including denial of service, user-to-root, remote-to-local, and probing. In order to evaluate the four proposed Hybrid NIDS models the NSL- KDD dataset are used, where 59586 records are randomly taken. All experiments have been performed using Intel Core 2 Duo 2.26 GHz processor with 2 GB of RAM with java implementation.

4.2 Performance Evaluation

The Comparison Criteria to evaluate the proposed network intrusion detection system are:

1. The speed of the ID system
2. The classification Accuracy

The classification accuracy of an intrusion detection system is measured by the *precision*, *recall* and $F - measure$; which are calculated based on the confusion matrix given in Table 4.1. The confusion matrix shows the four possible prediction outcomes; True negatives (TN), True positives (TP), False positives (FP) and False negatives (FN) [53, 8].

Table 4.1: Confusion Matrix

		Predicted Class	
		Normal	Attake
Actual Class	Normal	True positives (TP)	False positives (FP)
	Attake	False negatives (FN)	True negatives (TN)

where:

True negatives (TN): indicates the number of normal events are successfully labeled as normal.

False positives (FP): refer to the number of normal events being predicted as attacks.

False negatives (FN): The number of attack events are incorrectly predicted as normal.

True positives (TP): The number of attack events are correctly predicted as attack.

$$Recall = \frac{TP}{TP + FN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{4.3}$$

An IDS should achieve a high recall without loss of precision, where F-measure is a weighted mean that assesses the trade-off between them.

4.3 PCA-SVM ID Model

4.3.1 Case one: Compare the evolution performance of KDD'99 and NSL-KDD dataset.

We applied the SVM intrusion detection system on two test sets; the original KDD'99 test set (KDDTest) and NSL-KDD test set (KDDTest+).

Tables 4.2, the redundant records in the original KDD'99 cause the learning algorithms to be biased towards the frequent records (DOS and Prob attacks). Thus prevent the detection algorithms from learning unfrequented records (U2R and R2L attacks).These unbalanced distribution of the KDD'99 testing dataset completely affects the evaluation of the detection algorithms;5.1% for U2R and 70.2 for R2L . While, NSL-KDD test sets have no redundant records; hence, the performances of the learners are not biased by frequent records.

Table 4.2: KDD'99 and NSL-KDD dataset testing accuracy comparison

Class name	Original KDD'99 test set	NSL-KDD test set
	Test Accuracy	Test Accuracy
Normal	99.8%	99.5%
DoS	92.5%	97.5%
U2R	5.1%	86.6%
R2L	70.2%	81.3%
Probe	98.3%	92.8%

4.3.2 Case two: Evaluate the proposed SVM-PCA ID model NSL- KDD dataset

We have randomly taken 10776 training records and 7970 test records. The performance of the proposed PCA-SVM ID model includes testing accuracy and the processing speed during testing. The experimental results are shown in Table 4.3 and 4.4.

Table 4.3: SVM and PCA-SVM model Testing accuracy comparison

Class name	SVM system with 41-dimension feature Test Accuracy	SVM system with 23-dimension feature Test Accuracy
Normal	99.8%	99.5%
DoS	97.5%	99.9%
U2R	86.6%	81.2%
R2L	81.3%	54.6%
Probe	92.8%	95.3%

Table 4.3 shows the accuracy achieved for SVMs using full dimension data (without PCA) and after the features reduction (with PCA). The testing accuracies indicate that PCA can be used to reduce data dimension without sacrificing much performance in accuracy.

Table 4.4: SVM and PCA-SVM model Time speed comparison

	Train time (ms)	Test time (ms)
SVM system with 41-dimension feature	4.5	3.1
SVM system with 23-dimension feature	1.7	1.3

Table 4.4, illustrate that SVMs system with PCA will improve training and testing speed, which is important for real time network applications. It

is clear that the proposed SVMs system with PCA faster in training and testing than SVMs without PCA.

4.3.3 Conclusion

We test the new dataset NSL-KDD which solve important issues of KDD'99 dataset. The experiments show that, NSL-KDD dataset can be applied as an effective benchmark dataset to help researchers compare different intrusion detection models. We proposed a PCA feature selected SVM intrusion detection system. PCA algorithm was used in order to select a best subset of features for classifying. We build SVM system to evaluate the selected subset. We developed a series of experiments on NSL-KDD dataset to examine the effectiveness of our building IDS. The experiment results show that our system is able to speed up the training and testing process of intrusions detection which is important for high-speed network applications.

4.4 DBN-SVM ID Model

4.4.1 Case 1: DBN vs. SVM vs. Hybrid DBN-SVM model

A comparison between SVM, DBN and the proposed DBN-SVM model is shown in Table 4.5. The classification accuracy achieved using DBN as dimensional reduction method before SVM is improved than using SVM or DBN as a separate classifier. Also the testing speed of DBN-SVM model is improved which is important for real time network applications.

Table 4.5: SVM, DBN and hybride DBN-SVM model testing accuracy and speed comparison

Training percentage	SVM	DBN	DBN-SVM
20%	82.30	89.63	90.06
	(10.4 Sec)	(0.31 Sec)	(2.54Sec)
30% 87.6	89.44	91.50	
	(10.4 Sec)	(0.26 Sec)	(2.54Sec)
40% 88.33	89.54	92.84	
	(16.67Sec)	(0.24 Sec)	(3.07 Sec)

4.4.2 Case 2: DBN as feature reduction vs. different feature reduction methods

We compared the DBN as a feature reduction method with other well known feature reduction methods like PCA, Gain Ratio and chi square. Using DBN, PCA, Gain Ratio and chi square the 41 features of the NSL- KDD dataset is reduced to 13 features. Table 4.6 gives the testing performance accuracy of the reduced data using SVM classifier. Table 4.6 illustrate that DBN gives better performance than the other reduction methods.

Table 4.6: Performance accuracy of DBN with different feature reduction methods

Training percentage	PCA	Gain-Ratio	Chi-Square	DBN
20%	68.72	65.83	66.0	90.06
30%	68.98	65.88	65.68	91.50
40%	71.01	70.99	65.82	92.84

4.4.3 Conclusion

Deep Belief network has proved to be a good addition to the field of network intrusion classification. In comparison with known classifier and feature reduction methods, DBN provides a good result as a separate classifier and as a feature reduction method. Therefor, we proposed a hybrid DBN-SVM intrusion detection model, where DBN is used as a feature reduction method and SVM as a classifier. We examine the performance of the proposed DBN-SVM model by reducing the 41-dimensional of NSL-KDD dataset to approximately 87% of its original size and then classify the reduced data by SVM. The DBN-SVM model shows higher percentage of classification than SVM and enhances the testing time due to data dimensions reduction. Also, we compare the performance of the DBN as a feature reduction method with PCA, Gain Ratio and Chi-Square feature reduction methods.

4.5 GA-HNB ID Model

4.5.1 Case 1: Evaluate the classification performance and time speed of the GA-HNB ID Model

The classification performance measurements of HNB and GA-HNB ID model are given in Table 4.7 and 4.8 respectively. Table 4.7 shows the accuracy measurements achieved for HNB using full dimension data (41 features). While, Table 4.8 gives the accuracy measurements for the proposed hybrid GA-HNB anomaly intrusion detection model with 17 dimension feature.

Table 4.7: HNB accuracy measurements (41-dimension feature)

Class name	TP Rate	FP Rate	Precision	Recall	F-Measure
Normal	0.88	0.008	0.96	0.88	0.92
DoS	0.99	0.003	0.99	0.99	0.99
U2R	0.99	0.002	0.98	0.99	0.98
R2L	0.97	0.010	0.93	0.97	0.95
Probe	0.99	0.013	0.95	0.99	0.97

The testing speed of hybrid GA-HNB anomaly intrusion detection model is measured and given in table 4.9. The GA-HNB ID model improves the timing speed to 0.26 sec. which is very important for real time network applications. Also, the classification accuracy achieved using hybrid GA-

Table 4.8: GA-HNB accuracy measurements (17-dimension feature)

Class name	TP Rate	FP Rate	Precision	Recall	F-Measure
Normal	0.98	0.008	0.989	0.98	0.984
DoS	0.99	0.003	0.995	0.995	0.995
U2R	0.98	0.001	0.982	0.981	0.981
R2L	0.97	0.005	0.913	0.970	0.941
Probe	0.99	0.002	0.985	0.991	0.988

HNB ID model is improved than using HNB as a standalone classifier.

Table 4.9: HNB and GA-HNB model Timing and Testing accuracy comparison

	Time to build model (sec)	Test accuracy
HNB model with 41-dimension feature	1.12	97.10 %
GA-HNB model with 17-dimension feature	0.26	98.63%

4.5.2 Case 2: Compare Different feature selection methods Performance accuracy by HNB classifier

Various well known filter and wrapper feature selection methods are applied to NSL-KDD dataset. The effectiveness of these methods is evaluated by HNB classifier. Table 4.10 gives the comparison results based on 10 fold cross-validation.

Table 4.10: Different feature selection methods Performance accuracy with HNB classifier

feature selection method	Precision	Recall	F-Measure	overall accuracy %
Best first (10 features)	0.962	0.959	0.960	95.93
Greedy stepwise (11 features)	0.961	0.959	0.959	95.92
chi square (12 features)	0.982	0.982	0.982	98.19
Gain ratio (15 features)	0.957	0.954	0.954	95.40
Genetic (17 features)	0.987	0.986	0.986	98.63
PCA (25 features)	0.972	0.972	0.972	97.18

Figures 4.1 - 4.4 shows the $Precision$, $Recall$, $F - Measure$ and $Overallaccuracy$ of the hybrid anomaly GA-HNB ID model versus different feature selection methods.

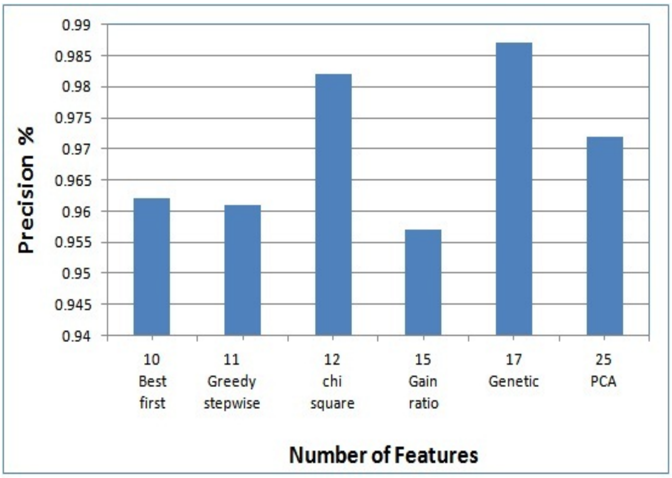


Figure 4.1: Precision of hybrid GA-HNB intrusion detection model

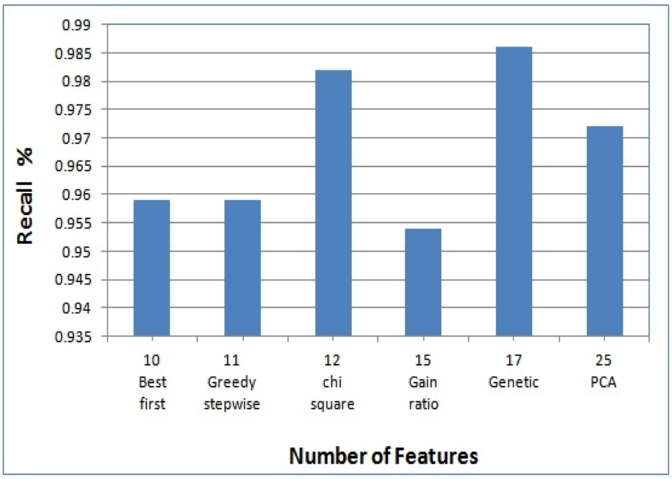


Figure 4.2: Recall of hybrid GA-HNB intrusion detection model

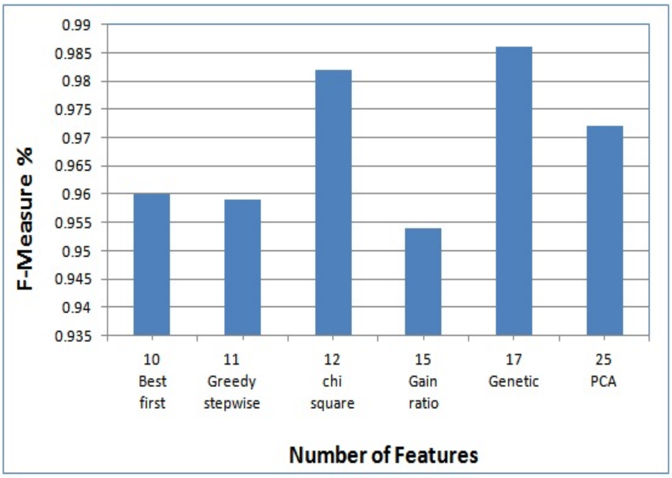


Figure 4.3: F-Measure of hybrid GA-HNB intrusion detection model

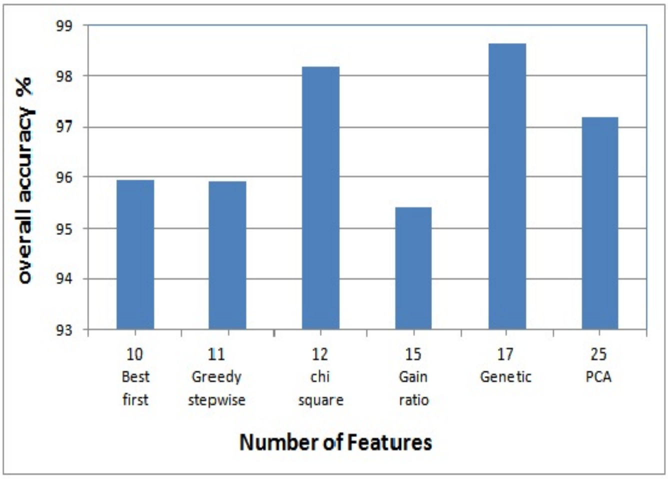


Figure 4.4: Overall accuracy of hybrid GA-HNB intrusion detection model

4.5.3 Conclusion

We proposed a hybrid Genetic algorithm and HNB anomaly intrusion detection approach, where genetic algorithm is used as a feature selection and then classify the reduced data by HNB classifier. The proposed hybrid anomaly GA-HNB ID model reduced the 41-dimensional of NSL-KDD dataset by 58.5% of its original size. GA-HNB ID model shows high percentage of classification 98.63% and enhances the testing speed to 0.26 sec. due to data dimension reduction (17 features). Also, we compare the performance of the GA as a feature selection method with different filter and wrapper feature selection method as PCA, Best first, Greedy stepwise, Gain Ratio and Chi-Square.

4.6 Hybrid Real-Time Discretize ID Model

We evaluate the proposed hybrid real-time Discretize framework on different categories of classifiers; tree classifiers (REPTree, C 4.5, Random Forest), rule based classifiers (Ridor, Decision table) and Naïve bayes classifier.

Table 4.11 and 4.12 shows the F-measures and speed achieved for the different set of classifiers; without applying any preprocessing phase, applying IEM discritization and finally applying IEM discritization combined with GA feature selection (14 features). The comparison results are based on 10 fold cross-validation.

Table 4.11: Comparison of F-measures and speed for tree classifiers

	REPTree		C4.5		Random Forest	
Preprocess approach	F-measure	speed (sec.)	F-measure	speed(sec.)	F-measure	speed(sec.)
Non	98.3%	6.07	98.8%	43.46	99.2%	34.58
Discretization	98.1%	3.75	99.0%	3.05	99.1%	2.87
Discretization + GA	98.7%	1.20	98.8%	0.77	99.3%	1.76

From table 4.11, applying IEM Discretization method leads to highly improve the speed of the systems especially for C4.5 classifier; which is very important for real time network intrusion detection systems. Also, the classification accuracy for REPTree and Random Forest classifier does not effect by discretization, while it is improved for C4.5 classifier. The systems speed shows another improvement when combining the discretization with GA faeture selection; which reduces the NSL-KDD dimentions from 41 features to 14 features.

Table 4.12: Comparison of F-measurs and speed for Rules based and Naïve bayes classifiers

	Ridor		Decision table		Naïve Bayes	
Preprocess approach	F-measure	speed(sec.)	F-measure	speed(sec.)	F-measure	speed(sec.)
Non	98.3%	435.57	96.4%	136.6	72.16%	4.21
Discretization	97.2%	129.16	96.3%	132.0	93.6%	0.21
Discretization + GA	97.8%	61.14	97.9%	25.5	94.5%	0.09

Table 4.12, gives the impact of applying discretization and applying discretization combined with GA feature selection. For Ridor classifier the

system speed shows a good improvement. Also, it is clear that, discretization has a positive impact on the naïve bayes classifier, that is, it helps to highly improve the detection accuracy and speed.

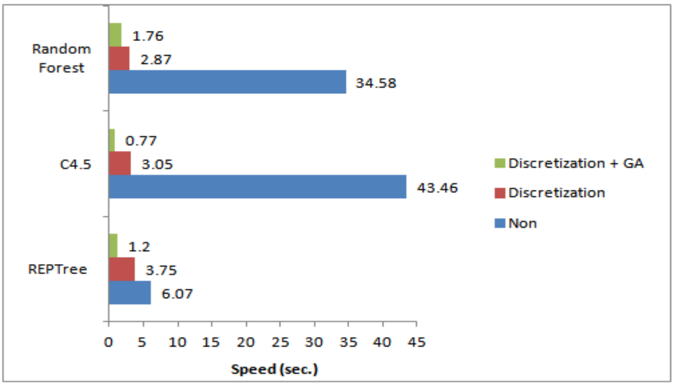


Figure 4.5: Speed comparision of the proposed network ID framework on tree classifiers.

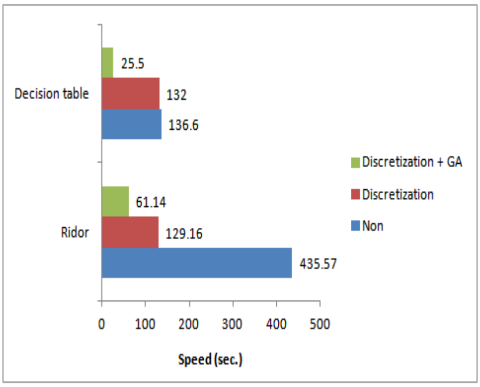


Figure 4.6: Speed comparision of the proposed network ID framework on Rule based classifiers.

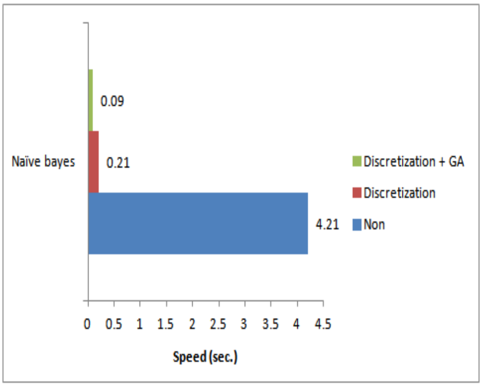


Figure 4.7: Speed comparision of the proposed network ID framework on NB classifiers.

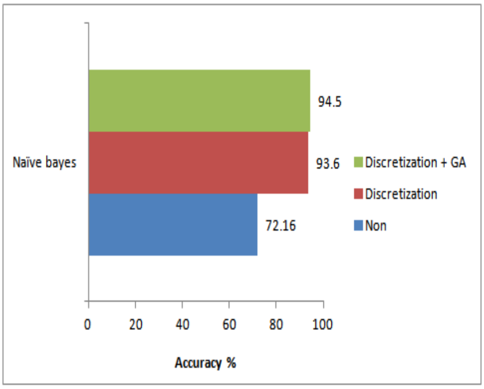


Figure 4.8: Accuracy comparision of the proposed network ID framework on NB classifiers.

Figures 4.5 - 4.7 shows the speed comparision of the proposed network ID framework for the different set of classifiers. Also, Figure 4.8 gives the accuracy comparision for the Naïve bayes classifier.

4.7 Conclusion

In these experiments, a real time discritize network ID framework is proposed. We explore the impact of applying IEM discretization and GA feature selection on the performance of network IDS. Different classifiers algorithms; rules based classifiers (Ridor, Decision table), trees classifiers (REPTree, C4.5, Random Forest) and Naïve bays classifier are used to evaluate the classification time and accuracy of the introduced network ID framework. Experiments on the NSL-KDD dataset show that IEM discretization helps to highly improve the time to classify the test instances. Which is an important factor for real time network IDS. Also, IEM discretization has a positive impact on the classification accuracy, especially for the naïve bayes classifier.

4.8 Comparison of The Proposed Hybrid NID Models

Table 4.13 shows the comparison results of the four proposed Hybrid Netwrok ID models.

Table 4.13: Comparison of of The Proposed Hybrid NID Models

Preprocess approach	Classifier	Hybrid Model	Features Number	F-measure (%)	Speed(sec.)
$^*FS$: PCA	SVM	PCA-SVM	23	87.1	1.7
FS:DBN	SVM	DBN-SVM	13	92.8	3.07
FS: GA	HNB	GA-HNB	17	98.6	0.26
FS:GA + $^*Disc$:IEM	Tree based	GA-IEM-C 4.5	14	98.8	0.77
FS:GA + Disc:IEM	Rule based	GA-IEM-DT	14	97.9	25.5
FS:GA + Disc:IEM	NB	GA-IEM-NB	14	94.5	0.09

$^*FS$: Feature Selection. $^*Disc$: Discritization.

From Table 4.13 for building an efficient NIDS; different classifiers algorithms and data preprocessing (Feature Selection and Discritization) are applied. Experiments on the NSL-KDD dataset show that the hybrid GA-IEM-C 4.5 model gives the best detection accuracy (98.8%) with speed 0.77 sec. While the hybrid GA-HNB model gives the best detection speed 0.26 sec which is very important for real time network applications. The hybrid GA-HNB model does not scarify much performance in accuracy (98.6%) than the C 4.5-GA-IEM model. Thus, HNB classifier hybrid with GA as a feture selection methods has proved to be a good addition to the field of network intrusion classification.

CHAPTER 5

Bi-Layer Behavioral-based Feature Selection Approach

Chapter 5

Bi-Layer Behavioral-based Feature Selection Approach

5.1 Introduction

Feature selection (FS) is a preprocessing step to machine learning, leads to increase the classification accuracy and reduce its complexity. FS methods are classified into tow main categories: filter and wrapper. Filter methods evaluate features without involving any learning algorithm, while wrapper methods depend on a learning algorithm for feature evaluation. Variety hybrid filter and wrapper methods have been proposed in the literature. However, hybrid filter and wrapper approaches suffer from the problem of determining the cut-off point of the ranked features. This leads to decrease the classification accuracy by eliminating important features. In this chapter we proposed a Hybrid Bi-Layer behavioral-based feature selection approach, which combines filter and wrapper feature selection methods. The proposed approach solves the cut-off point problem for the ranked features. It consists of two layers, at the first layer Information gain is used to rank the features and select a new set of features depending on a global maxima classification accuracy. Then, at the second layer a new subset of features is selected from within the first layer redacted data set by searching for a group of local maximum classification accuracy. To evaluate the proposed approach it is applied on NSL-KDD dataset.

5.2 Proposed Approach: Bi-layer Behavioral-Based Feature Selection

In network intrusion detection problem, the number of selected features to train the IDS are still high after applying the conventional feature selection methods. Thus, a second feature selection layer should be added to decrease the number of features without affecting the classification accuracy.

The Bi-layer behavioral-based feature selection approach Hybrids filter and wrapper feature selection methods. It consists of two layers as shown in Figure 5.1. In the first layer information gain is used to rank the features, where the classification accuracy is tested sequentially starting from the top ranked features. Depending on a global maxima classification accuracy a new set of features is selected. Then, the second layer select a new set of features from within the first layer redacted data in order to increase the number of reduced features. It search for a group of local maximum classification accuracy depending on the variation of the classification accuracy.

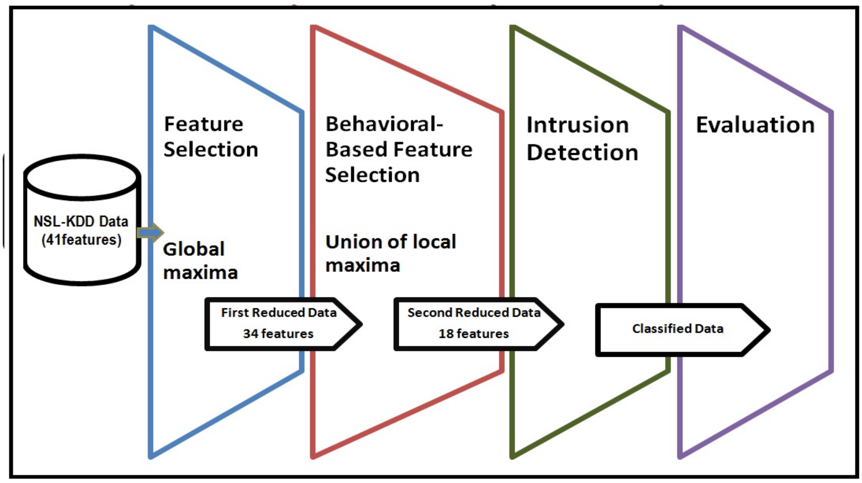


Figure 5.1: Bi-layer behavioral-based feature selection approach

The proposed Bi-layer behavioral-based feature selection approach is based on a hypothesis that features that leads to classification accuracy less than or equal to the accuracy of previous ranked feature can be removed. Figure 5.2 shows the variation of the classification accuracy, where a global maxima accuracy is defined; within the global maxima area a set of local maxima is defined. Thus, the new features set will be the union of the shaded area under the locals maxima and global maxima.

The algorithm of the proposed Bi-layer behavioral-based feature selection approach is given in algorithm 9:

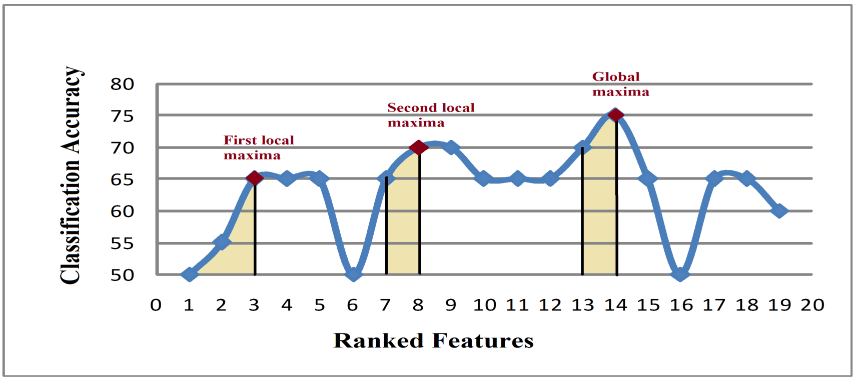


Figure 5.2: The variation of the classification accuracy on ranked features

Algorithm 9 Bi-layer behavioral-based feature selection approach

```
Input: set $A$ of $n$ selected features, sorted according the information gain ranked values.
Output: reduced set $R$ of the selected features

  Define an array $CA$ of $n$ cells
  for $i = 1$ to $n$ do
    Construct a data set that contains the first $i$ features in the input data set
    Apply J48 classifier and assign the resulted accuracy to $CA$[i]
  end for

  Define a variable $c$ to indicate the the current classification accuracy
  Define a variable $p$ to indicate the the previous classification accuracy
  Define a variable $max_c$ to indicate the maximum accuracy reached so far
  $max_c = 0$
  $p = 0;$
  for $i = 1$ to $n$ do
    $c = CA$[i]
    if $c > p$ and $c > max_c$ then
      $A$[i] to the set $R$
      $max_c = c$
    end if
    $p = c$
  end for
```

5.3 Experimental work and Analysis

The NSL- KDD dataset are taken to evaluate the proposed Bi-layer behavioral-based feature selection approach. All experiments have been performed using Intel Core 2 Duo 2.26 GHz processor with 2 GB of RAM and weka software [120].

5.3.1 Case 1: Evaluation of the Bi-layer behavioral-based feature selection approach

The 41 features of the NSL-KDD data set are evaluated and ranked according to the information gain method. Then, forward feature selection is applied to the ranked feature space, where classification accuracy is measured by j48 classifier. The variation of j48 classification accuracy is given in figure 5.3, as shown the classification accuracy leads to seven local maxima and a global maxima. In the conventional forward feature selection method all the 35 features before the global maxima will be selected. However, in the proposed Bi-layer behavioral-based feature selection approach, only 20 features will be selected depending on the local maxima points.

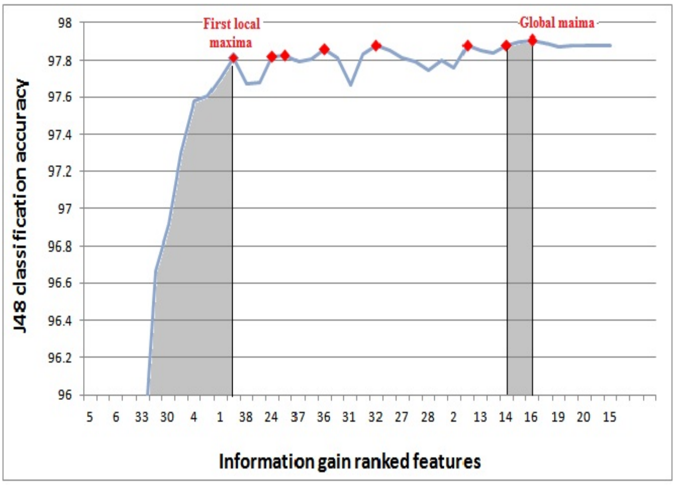


Figure 5.3: Variation of J48 classification accuracy on ranked feature space

Table 5.1 gives the $F - Measure$ comparison results based on 10 fold cross-validation.

Table 5.1: $F - Measure$ comparison of the proposed Bi-layer behavioral-based feature selection and conventional forward feature selection

Feature selection approach	Number of features	F-Measure
Non	41	97.9%
forward feature	35	98.6%
Bi-layer behavioral-based	20	99.2%

It is clear from table 5.1 that, for the Bi-layer behavioral-based feature selection the classification accuracy increased to 99.2% while the number of feature decreased to 20 features.

Table 5.2: Timing and Testing accuracy comparison of the proposed Bi-layer behavioral-based feature selection approch

	Time to build model (sec)	Test accuracy
J48	36.15	97.9%
J48 + Bi-layer behavioral-based	24.43	99.2 %

Table 5.2 gives the timing speed of building the proposed hybrid anomaly intrusion detection model (J48 + Bi-layer behavioral-based). From Table 5.2 it is clear that timing speed of the hybrid J48 + Bi-layer behavioral-based is improved which is very important for real time network applications. Also, the classification accuracy achieved, based on 10-fold cross-validation, using the proposed hybrid anomaly intrusion detection approach is improved than using J48 as a standalone classifier.

5.3.2 Case 2: Bi-layer behavioral-based feature selection vs. different feature selection approaches

Bi-layer behavioral-based feature selection approach is compared with various well known feature selection approaches as PCA, Gain Ratio, chi square and information gain. Based on 10 fold cross-validation Table 5.3 gives the F-Measure accuracy of the reduced data using Bi-layer behavioral-based feature selection and the other well known feature selection approaches.

Table 5.3: $F - Measure$ comparison of the proposed Bi-layer behavioral-based feature selection with other feature selection approaches

Feature selection approach	Number of features	F-Measure
PCA	25	97.6%
Gain-Ratio	34	98.8%
Chi-Square	28	98.8%
Information Gain	35	98.6%
Bi-layer behavioral-based	20	99.2%

From Figure 5.4, it is clear that the F-measure for the Bi-layer behavioral-based approach shows better result when compared to the other approaches.

5.3.3 Chapter Conclusion

This chapter proposed a Bi-Layer behavioral-based feature selection approach that depends on the behavior of the classification accuracy according to ranked feature. The proposed Bi-Layer behavioral-based feature selection approach demonstrate the superiority over well known feature selection approaches. The experiments shows that the proposed approach improved the accuracy to 99.2%, while reducing the number of features from 41 to 20 features.

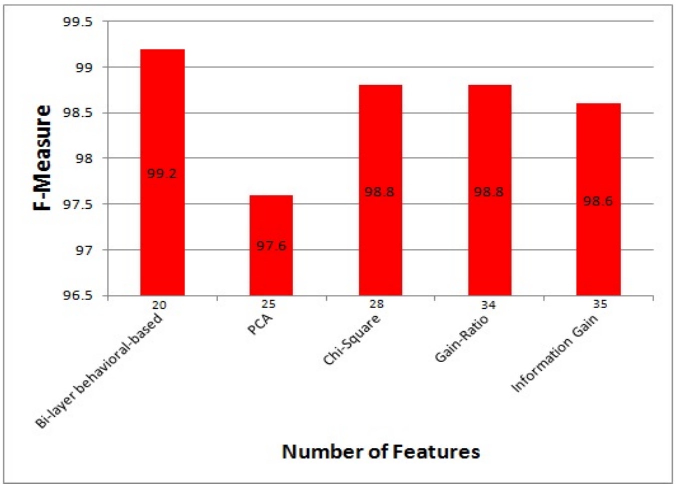
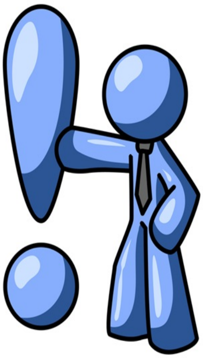


Figure 5.4: F-Measure of proposed Bi-layer behavioral-based feature selection vs different feature selection approaches

CHAPTER 6

Conclusions & Future Directions



Chapter 6

Conclusion and Future Directions

6.1 Conclusions:

The thesis objectives is to outline the idea of network intruder discovery in IDS. The thesis presents a solution to the problem of data dimension and detection of network intrusions. The solution described in this thesis can be summarized into two direction:

1. Proposing four hybrid anomlay Network intrusion detection models.
2. Proposing a new feature selection approach "Bi-Layer behavioral-based feature selection approach".

I. The four proposed Hybrid NIDS models described in chapter 3 has the advantages of enhance the detection accuracy and testing speed by reducing the feature dimension space. The four proposed anomaly Network intrusion detection models is validate through the following: (1)

Apply the proposed hybrid model on NSL-KDD intrusion dataset, (2) compare the propose hybrid model with other well known feature selection techniques. Where the experimental results described in chapter 4 shows that:

1) PCA-SVM model gives detection accuracy 87.1% with speed 1.7 sec.,
2) DBN-SVM model gives detection accuracy 92.8% with speed 3.07sec.,
3) GA-HNB model gives detection accuracy 98.6% with speed 0.26 sec.,
4) and GA-IEM-C4.5 model gives detection accuracy 98.8% with speed 0.77 sec.

from the results obtained it is clear that the hybrid C 4.5-GA-IEM model gives the best detection accuracy (98.8%) with speed 0.77 sec. However, the hybrid GA-HNB model gives the best detection speed 0.26 sec without sacrificing much performance in accuracy (98.6%) than the GA-IEM-C 4.5 model. The detection speed is a very important factor for real time network applications. Thus, HNB classifier hybrid with GA as a feture selection methods has proved to be a good addition to the field of network intrusion classification.

II. The Bi-Layer behavioral-based feature selection approach is proposed in chapter 5 which:

- combines filter and wrapper feature selection methods.
- solves the cut-off point problem for the ranked intrusion features.

The Experiments on well known NSL-KDD datasets are demonstrate the efficiency of the proposed Bi-Layer behavioral-based feature selection approach. Where, the detection accuracy is decreased to 99.2% by reducing the number of features from 41 to 20 features.

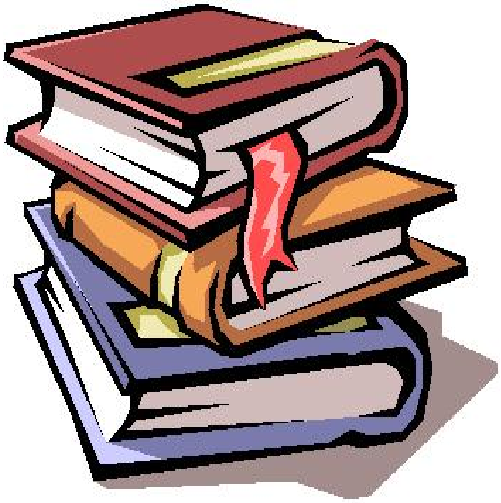
6.2 Future Directions:

The intrusion detection problem has three basic competing requirements: (1) speed, (2) accuracy and (3) adaptability. The four proposed NIDS model and the new proposed Bi-Layer behavioral-based feature selection approach solves the detection speed and detection accuracy problem. The model resulting high detection speed and accuracy represents a quality of service issue.

However, network intrusion detection systems are still reliant on human input in order to maintain the accuracy of the system. In case of misuse-based NID systems, security experts examine the attacks to create corresponding detection signatures. Also, in the case of anomaly-based NID systems, security experts are needed to define the normal behavior. This leads to the adaptability problem. Thus, the capability of the current network intrusion detection systems for adaptation is very limited, since human intervention is always required. This makes them inefficient for adapting to the changes of the complex nature of network environments. Although a new research area, incorporation of machine learning algorithms provides a potential solution for accuracy and adaptability of the intrusion detection problem.

The human immune system provides the human body with a high level of protection from viruses, in a self-organized and distributed manner. Thus, the biologically inspired approaches in NIDS area, including the use of immune-based systems will be able to meet the adaptation challenge. An immune system is autonomous which owns features of distribution and agents. Therefor, for solving the NIDS adaptation problem, we are looking for proposing an artificial immune agent system for network intrusion detection. Where, the artificial immune-agent NIDS will benefites from the immune system self-adaptation and the common features inherited from the general agent.

References



References

[1] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection", Information Sciences, vol.177, pp. 3799-3821, 2007.

[2] N. I. of Standards & Technology, "An Introduction to Computer Security: The NIST Handbook", NIST, Ed. U.S. Department of Commerce, 2006.

[3] V. Marinova-Boncheva, "A Short Survey of Intrusion Detection Systems", Bulgarian Academy Of Sciences, Problems Of Engineering Cybernetics And Robotics, 58, ISSN 0204-9848, 2007.

[4] Rick Lehtinen, Deborah Russell and G. T. Gangemi, "Computer Security Basics", 2nd edition,OReilly Media, 2006.

[5] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "The architecture of a network level intrusion detection system, Technical report,

Computer Science Department, University of New Mexico, August 1990.

[6] J.P Anderson, “Computer Security Threat Monitoring and Surveillance, Technical report, James P Anderson Co., Fort Washington, Pennsylvania, April 1980.

[7] W. Stallings, “Cryptography and network security principles and practices”, USA: Prentice Hall, 2006.

[8] S. Wu and W. Banzhaf, “The use of computational intelligence in intrusion detection systems: A review”. Applied Soft Computing, vol. 10, pp. 1-35, 2010.

[9] G. Wang , J. Hao , J. Ma and L. Huang, “A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering”. Expert Systems with Applications, vol. 37, pp. 6225-6232, 2010.

[10] S. Horng , M. Su , Y. Chen , T. Kao , R. Chen , J. Lai and C. Perkasa, “A novel intrusion detection system based on hierarchical clustering and support vector machines”. Expert Systems with Applications, vol. 38, pp. 306-313, 2011.

[11] M. Dash, and H. Liu, “Feature selection for classifications”. Intelligent Data Analysis: An International Journal, vol. 1, pp. 131-156, 1997.

[12] G. Kuchimanchi, V. Phoha, K. Balagani and S. Gaddam,“Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems”, Information Assurance Workshop, In Proceedings of the Fifth Annual IEEE SMC,NY, USA pp.195-202, 2004.

[13] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai and K. Dai, “An efficient intrusion detection system based on support vector machines and gradually feature removal method”, Expert Systems with Applications, vol. 39, pp. 424-430, 2012

[14] F. Amiri, M. Yousefi, C. Lucas, A. Shakery and N. Yazdani, “Mutual information-based feature selection for intrusion detection systems”, Journal of Network and Computer Applications, vol.34, pp.1184-1199, 2011.

[15] P. Pudil, J. Novovicov and J. Kittler, “Floating search methods in feature selection,” Pattern Recognition Letters, vol. 5, pp. 1119-1125, 1994.

[16] M. Dash and H. Liu, “Consistency based search in feature selection,” Artificial Intelligence, vol. 151, pp. 155176, 2003.

[17] S. Park, M. Shazzad and S. Kim, "Towards modeling lightweight intrusion detection system through correlation-based hybrid feature selection," In Proceedings of of the First SKLOIS conference on Information Security and Cryptology,Beijing, China, pp. 279-289, 2005.

[18] I. Gheyas and L. Smith, "Feature subset selection in large dimensionality domains," Pattern Recognition, vol. 43, pp. 513, 2010.

[19] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection.", Journal of Machine Learning Research vol. 3, pp. 1157-1182, 2003.

[20] X. Xu, "Adaptive Intrusion Detection Based on Machine Learning: Feature Extraction, Classifier Construction and Sequential Pattern Prediction". International Journal of Web Services Practices, vol. 2, pp. 49-58, 2006.

[21] M. Sheikhan and A. Shabani, "Fast Neural Intrusion Detection System Based on Hidden Weight Optimization Algorithm and Feature Selection.", World Applied Sciences Journal, vol.7, pp. 45-53, 2009.

[22] Z. Sun , G. Bebis and R. Miller, "Object detection using feature subset selection". Pattern Recognition, vol. 37, pp. 2165-2176, 2004.

[23] G. Stein , B. Chen , A. Wu and K. Hua, “decision tree classifier for network intrusion detection with GA-based feature selection”. In Proceedings of 43rd annual Southeast regional conference, New York, USA, vol.2, pp. 136-141, 2005.

[24] H. F. Eid , A. Darwish, A. and A. Abraham, “Principle Components Analysis and Support Vector Machine Based Intrusion Detection System”, The 10th IEEE international conference in Intelligent Design and Application (ISDA2010) 29 November - 1 December, Cairo Egypt, pp.363-367, 2010..

[25] M. Salama, H. F. Eid , A. Darwish and A. Hassanien, “Hybrid Intelligent Intrusion Detection Scheme”, 15th Online World Conference on Soft Computing in Industrial Applications, 15th to 27th November, Springer in ”Advances In Intelligent and Soft Computing, pp. 295-302, 2010.

[26] H. F. Eid, A. Darwish, A. Hassanien and T. Kim, “Intelligent Hybrid Anomaly Network Intrusion Detection System”, International Conference on Future Generation Communication and Networking. CCIS/ LNCS series Springer, (Indexed by SCOPUS, EI) December 8-10, 2011 in Jeju Grand Hotel, Jeju Island, Korea , , pp. 209-218, 2011.

[27] H. F. Eid and A. Hassanien, ”Improved Real-Time Discretize Network Intrusion Detection Model”, Seventh International Conference on Bio-Inspired Computing: Theories and Application (BIC-TA 2012), December 14- 16, Gwalior, India, pp.99-109, 2012.

[28] H. F. Eid, M. Salama, A. Hassanien and T. Kim, “Bi-Layer Behavioral-based Feature Selection Approach for Network Intrusion Classification”, The International Conference on Security Technology (SecTech 2011), CCIS/LNCS series Springer, (Indexed by SCOPUS, EI) December 8 - 10, 2011 in Jeju Grand Hotel, Jeju Island, Korea , pp. 195-203, 2011.

[29] K. Krishnakumar, “Intelligent systems for aerospace engineering an overview, NASA Technical Report, Document ID:20030105746, 2003.

[30] T. A. Byrd and R. D. Hauser, “Expert systems in production and operations management: research directions in assessing overall impact, Int. J. Prod. Res., vol. 29, pp. 2471-2482, 1991.

[31] Y. Dote and S. J. Ovaska, “Industrial Applications of Soft Computing: A Review, Proceedings of the IEEE, vol. 89, pp. 1243-1265, 2001.

[32] R. Eberhart, P. Simpson, and R. Dobbins, “Computational Intelligence PC Tools”, Academic Press, Boston, 1996.

[33] D. Poole, A. Mackworth and R. Goebel, "Computational IntelligenceA Logical Approach", Oxford University Press, Oxford, UK, 1998.

[34] J. C. Bezdek, "What is computational intelligence? In: Computational Intelligence Imitating Life", pp. 112, IEEE Press, New York, 1994.

[35] I.J. Rudas, "Hybrid Systems (Integration of Neural Networks, Fuzzy Logic, Expert Systems, and Genetic Algorithms)", In: Encyclopedia of Information Systems, Academic Press, pp. 1141-1148, 2002.

[36] J. Jang, C. Sun and E. Mizutani, "Neuro-Fuzzy and Soft Computing". Prentice Hall, 1997.

[37] J. Gero and V. Kazakov, "Evoloving design genes in space layout problems". Artificial Intelligence in Engineering, vol. 12, pp. 163-176, 1998.

[38] J. Damsky and J. Gero, "An evolutionary approach to generating constraint-based space layout topologies". Kluwer Academic Publishing, pp. 855-874, 1997.

[39] J. Jo and J. Gero, “Space layout planning using an evolutionary approach”. Artificial Intelligence in Engineering, vol. 12, pp. 163-176, 1998.

[40] L. Caldas, “GENE ARCH: An evolution-based generative design system for sustainable architecture”. Lecture Notes in Artificial Intelligence Springer-Verlag Berlin Heidelberg, pp. 109-118, 2006.

[41] S. Bandaru and K. Deb, “Automated discovery of vital design knowledge from Pareto-optimal solutions: first results from engineering design”. IEEE Congress on Evolutionary Computation - CEC2010, Barcelona, Spain, 2010.

[42] D.B. Fogel, “What is evolutionary computation?”. IEEE Spectrum, vol. 37, pp. 28-32, 2000.

elberg, 2007, pp.113.

[43] V. Vapnik, “Statistical learning theory” New York: Wiley, 1998.

[44] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition”, Data Mining and Knowledge Discovery, vol. 2, pp.121-167, 1998.

[45] S. Kim, K. S. Shin and K. Park, “An application of support vector machines for customer churn analysis: Credit card case” Lecture Notes in Computer Science, vol. 3611, pp. 636-647, 2005.

[46] S. Kim, S. Yang and K. S. Seo, “Home photo categorization based on photographic region templates” Lecture Notes in Computer Science, vol. 3689, pp. 328-338, 2005.

[47] L. Jiang , H. Zhang and Z. Cai, “A Novel Bayes Model: Hidden Naive Bayes”. IEEE Tran. on Knowledge and Data Engineering, vol. 2, pp. 1361-1371, 2009.

[48] A. K. Noulas and B.J.A. Krse, “Deep Belief Networks for Dimensionality Reduction”, Belgian-Dutch Conference on Artificial Intelligence, Netherland, 2008.

[49] H.Larochelle and Y.Bengio, “Classification using discriminative restricted boltzmann machines”, In Proceedings of the 25th international conference on Machine learning,Helsinki, Finland, vol. 307, pp. 536-543, 2008.

[50] L. McAfee, ”Document Classification using Deep Belief Nets”, CS224n, Sprint 2008.

[51] H. Larochelle, Y. Bengio, J. Louradour and P. Lamblin, “Exploring Strategies for Training Deep Neural Networks”, Journal of Machine Learning Research, vol.10, pp.1-40, 2009.

[52] J. Holland, “Adaptation in Natural and Artificial Systems”. University of Michigan Press, Ann Arbor, MI, 1975.

[53] R. Duda , P. Hart and D. Stork, “Pattern Classification”, JohnWiley & Sons, USA, 2nd edition, 2001.

[54] B. Jiang , X. Ding , L. Ma , Y. He , T. Wang and W. Xie, “A Hybrid Feature Selection Algorithm:Combination of Symmetrical Uncertainty and Genetic Algorithms”. The Second International Symposium on Optimization and Systems Biology OSB’08), China, pp. 152-157, 2008.

[55] L. Davis, “Handbook of Genetic Algorithms”, Van Nostrand Reinhold, New York, 1991.

[56] W. Martin and W. Spears, “Foundations of Genetic Algorithms”, Morgan Kaufmann, San Francisco, CA, 2001.

[57] R. Eberhart , J. Kennedy, “A new optimizer using particle swarm theory”, In Proceeding of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp.39-43, 1995.

[58] G. Venter and J. Sobieszczanski-Sobieski, “Particle Swarm Optimization,” AIAA Journal, vol. 41, pp. 1583-1589, 2003.

[59] Y. Liu, G. Wang, H. Chen, and H. Dong, “An improved particle swarm optimization for feature selection”, Journal of Bionic Engineering, vol.8, pp.191-200, 2011.

[60] L. Zadeh, “Fuzzy sets.” Information and Control, vol. 8, pp. 338352, 1965.

[61] L. Zadeh , “Fuzzy logic and approximate reasoning.” Synthese, vol. 30, pp. 407-428, 1975.

[62] J. Mendel. “Fuzzy logic systems for engineering: a tutorial.” Proceedings of the IEEE, vol. 83, pp. 345-377, 1995.

[63] C. Shang and Q. Shen. “Aiding classification of gene expression data with feature selection: a comparative study”. Computational Intelligence Research, vol. 1, pp. 68-76, 2006.

[64] A. Sung, S. Mukkamala, “Identifying important features for intrusion detection using support vector machines and neural networks”. In Proceedings of the Symposium on Applications and the Internet, pp. 209-216, 2003.

[65] M. Dash, K. Choi, P. Scheuermann and H. Liu, “Feature selection for clustering a filter solution”, In Proceedings of the Second International Conference on Data Mining,Washington, DC, USA, pp. 115-122, 2002.

[66] D. Koller and M. Sahami, “Toward optimal feature selection”, In Proceedings of the Thirteenth International Conference on Machine Learning,Bari, Italy, pp. 284-292, 1996.

[67] C. Tsang, S. Kwong and H. Wang, “Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection”, Pattern Recognition, vol. 40, pp. 2373-2391, 2007.

[68] L. Yu and H. Liu, “Feature selection for high-dimensional data: a fast correlation-based filter solution”, In Proceedings of the twentieth International Conference on Machine Learning,Washington DC,USA, pp. 856-863, 2003.

[69] Y. Kim, W. Street and F. Menczer, “Feature selection for unsupervised learning via evolutionary search”, In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, pp.365-369, 2000.

[70] R. Kohavi and G.H. John, “Wrappers for feature subset selection”, Artificial Intelligence, vol. 97, pp. 273-324 , 1997.

[71] X. Jin, A. Xu, R. Bie and P. Guo, “Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles” , Lecture Notes in Computer Science, 3916, pp. 106-115, 2006.

[72] M. Ben-Bassat, “Pattern recognition and reduction of dimensionality”, Handbook of Statistics II, vol. 1, pp. 773-791, North-Holland, Amsterdam, 1982.

[73] H. Peng, F. Long and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max- relevance ,and min redundancy”, IEEE Transactions on Pattern Analysis and Machine Intelligence 27, pp. 1226-1238, 2005.

[74] J. R Quinlan, “Induction of Decision Trees, Machine Learning, vol. 1, pp. 81-106, 1986.

[75] F. Jemili, M. Zaghdoud and M. Ahmed,“Intrusion detection based on Hybrid propagation in Bayesian Networks”, In proceedings of the IEEE international conference on Intelligence and security informatics, Dallas, Dallas, TX, pp. 137-142, 2009.

[76] Veerabhadrappa and L. Rangarajan, “Bi-level dimensionality reduction methods using feature selection and feature extraction”, International Journal of Computer Applications, vol. 4, pp. 33-38, 2010.

[77] E. Oja, “Principal components, minor components, and linear neural networks”. Neural Networks, vol. 5, pp. 927-935, 1992.

[78] K. Labib and V.R. Vemuri, “Detecting and visualizing denial-of-service and network probe attacks using principal component analysis” In Third Conference on Security and Network Architectures, La Londe, (France), 2004.

[79] M. Shyu, S. Chen, K. Sarinnapakorn and L. Chang, “A Novel Anomaly Detection Scheme Based on Principal Component Classifier” In Proceedings of ICDM’03, Melbourne, Florida, pp. 172-179, 2003.

[80] T. Mitchell. “Machine Learning”. McGraw-Hill, New York, USA, 1997.

[81] E. Sivasankar, H. Sridhar, V. Balakrishnan and K. Ashwin, “Comparison of dimensionality reduction techniques using a backpropagation neural network based classifier,” International Journal of Information Acquisition,8, pp. 161-169, 2011.

[82] W. Wang, S. Gombault and T. Guyet, “Towards fast detecting intrusions: using key attributes of network traffic, In Proceedings of the Third International Conference on Internet Monitoring and Protection, IEEE, Bucharest, pp.86-91, 2008.

[83] M. Mizianty, L. Kurgan and M. Ogiela, “Discretization as the enabling technique for the Naïve Bayes and semi-Naïve Bayes-based classification”, The Knowledge Engineering Review, vol. 25, pp. 421-449, 2010.

[84] S. Kotsiantis and D. Kanellopoulos, “Discretization Techniques: A recent survey”,GESTS International Transactions on Computer Science and Engineering, vol.32, pp. 47-58, 2006.

[85] H. Liu, F. Hussain, C. Tan and M. Dash, “Discretization: an enabling technique”, Data Mining and Knowledge Discovery, vol. 6, pp. 393-423, 2002.

[86] J. Dougherty, R. Kohavi and M. Sahami, “Supervised and unsupervised discretization of continuous features”, In Proceedings of the 12th international conference on machine learning, San Francisco: Morgan Kaufmann; pp. 194-202, 1995.

[87] H. Steck and T. Jaakkola, “Predictive discretization during model selection”, In Proceedings of DAGM Symposium In Pattern Recognition, Tbingen, Germany, pp. 1-8, 2004.

[88] R. Kerber, “Chimerge: discretization of numeric attributes”, In Proceedings of the 9th International Conference of Artificial Intelligence, Cambridge, UK, pp. 123-128, 1992.

[89] J. Cerquides and R. Lopez, “Proposal and Empirical Comparison of a Parallelizable Distance Based Discretization Method”. In Proceedings of the III International Conference on Knowledge Discovery and Data Mining (KDDM97). Newport Beach, California USA, pp. 139-142, 1997.

[90] U. Fayyad and K. Irani, “Multi-interval discretization of continuous-valued attributes for classification learning”, In Proceedings of the International Joint Conference on Uncertainty in AI. Morgan Kaufmann, San Francisco, CA, USA, pp. 1022-1027, 1993.

[91] H. Debar, M. Dacier and A. Wespi, “Towards a taxonomy of intrusion-detection systems” Computer Networks, vol. 31, pp. 805-822, 1999.

[92] S. Axelsson, “Intrusion detection systems: A survey and taxonomy”, Department of Computer Engineering, Chalmers University of Technology,Tech. Rep.,2000.

[93] C. Endorf, E. Schultz, J. Mellander, “Intrusion Detection & Prevention, McGraw-Hill,New York, USA, 2004.

[94] R. Bace and P. Mell, “Nist special publication on intrusion detection systems, National Institute of Standards and Technology, Tech. Rep., 2001.

[95] J. Sommers, V. Yegneswaran and P. Barford, “A framework for malicious workload generation, in 4th ACM SIGCOMM conference on Internet measurement. New York, NY, USA: ACM, pp. 82-87, 2004.

[96] E. Biermann, E. Cloete and L.M. Venter, “A comparison of intrusion detection Systems”, Computer and Security, vol. 20, pp. 676-683, 2001.

[97] T. Verwoerd and R. Hunt, “Intrusion detection techniques and approaches”, Computer Communications, vol. 25, pp.1356-1365, 2002.

[98] H. Zhengbing, L. Zhitang and W. Junqi, “A novel network intrusion detection system (nids) based on signatures search of data mining. In Proceedings of the 1st international conference on Forensic ap-

plications and techniques in telecommunications, information, and multimedia. ICST, Brussels, Belgium, pp. 1-7, 2008.

[99] K. Ilgun, R.A. Kemmerer and P.A. Porras, “State transition analysis:A rule-based intrusion detection approach” IEEE Trans. Software Eng. vol. 21, pp. 181-199, 1995.

[100] D. Marchette, “A statistical method for profiling network traffic”. In Proceedings of the First USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, CA. pp. 119-128, 1999.

[101] D. Denning, “An intrusion detection model. IEEE Transactions on Software Engineering, vol. 13, pp. 222-232, 1987.

[102] S. Mukkamala , G. Janoski and A. Sung, “Intrusion detection: support vector machines and neural networks”. In proceedings of the IEEE International Joint Conference on Neural Networks (ANNIE), St. Louis, MO, pp. 1702-1707, 2002.

[103] D. Brown, B. Suckow and T. Wang. “A survey of intrusion detection systems”,Department of computer. Science, University of California, USA, 2001.

[104] E. Lundin and E. Jonsson, "Anomaly-based intrusion detection: privacy concerns and other problems", Computer Networks, vol. 34, pp. 623-640, 2002.

[105] F. Gong, "Deciphering detection techniques: Part ii anomaly-based intrusion detection. White Paper, McAfee Network Security Technologies Group, March 2003.

[106] MIT Lincoln Laboratory, DARPA Intrusion Detection Evaluation, http://www.ll.mit.edu/CST.html,MA, USA. July, 2010.

[107] D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. Mcclung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation, In Proceedings of the 2000 DARPA Information Survivability Conference and Exposition,South Carolina, USA, pp. 12-26, 2000.

[108] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection. In Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection. Springer-Verlag, Pittsburgh, PA, USA, pp. 220-237, 2003.

[109] J. McHugh, “Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory”. In ACM Transactions on Information and Systems Security, vol. 3, 2000.

[110] R. Durst, T. Champion, B. Witten, E. Miller, and L. Spagnuolo, “Testing and evaluating computer intrusion detection systems. Commun. ACM, vol. 42, pp. 53-61, 1999.

[111] KDD’99 dataset, http://kdd.ics.uci.edu/databases, Irvine, CA, USA, July, 2010.

[112] H. G. Kayacik , A. N. Zincir-Heywood and M. I. Heywood, “Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets”, In Proceedings of the Third Annual Conference on Privacy, Security and Trust (PST-2005),New Brunswick, Canada, 2005.

[113] K. Leung and C. Leckie, “Unsupervised anomaly detection in network intrusion detection using clusters”, In Proceedings of the Twenty-eighth Australasian conference on Computer Science, vol. 38, pp. 333- 342, 2005.

[114] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set” In Proceeding of the 2009

IEEE symposium on computational Intelligence in security and defense application (CISDA), Ottawa, 2009.

[115] M. Poon, G. Hamarneh and R. Abugharbieh, "Efficient interactive 3D Livewire segmentation of complex objects with arbitrary topology", Computerized Medical Imaging and Graphics, vol. 32, pp. 639-650, 2008.

[116] J. E. Nam, M. Maurer and K. Mueller, "A high-dimensional feature clustering approach to support knowledge-assisted visualization", Computers & Graphics, vol. 33, pp. 607-615, 2009.

[117] H. Khotanlou, O. Colliot, J. Atif and I. Bloch, "3D brain tumor segmentation in MRI using fuzzy classification, symmetry analysis and spatially constrained deformable models", Fuzzy Sets and Systems, vol. 160, pp. 1457-1473, 2009.

[118] C. Tsai , Y. Hsu, C. Lin and W. Lin, "Intrusion detection by machine learning: A review", Expert Systems with Applications, vol. 36, pp.11994-12000, 2009.

[119] U. Fayyad and K. Irani, "Multi-interval discretization of continuousvalued attributes for classification learning". In Thirteenth International Joint Conference on Articial Intelligence, pp. 1022-1027, (1993).

[120] Weka: Data Mining Software in java http://www.cs.waikato.ac.nz/ml/weka/, 2012.