

Manuscript Number: KNOSYS-D-16-02076

Title: An Efficient Intrusion Detection System based on Hypergraph -
Genetic Algorithm for Parameter Optimization and Feature Selection in
Support Vector Machine

Article Type: Full Length Article

Keywords: Hypergraph; Genetic algorithm; Support vector machine; feature
subset; kernel parameters; intrusion detection systems.

Corresponding Author: Dr. Shankar Sriram V S, Ph.D

Corresponding Author's Institution: SASTRA University

First Author: Gauthama Raman M R, M.Tech., [Ph.D]

Order of Authors: Gauthama Raman M R, M.Tech., [Ph.D]; Nivethitha Somu,
M.Sc.,M.Tech., [Ph.D]; Kannan Kirthivasan, M.Sc., M.Ed.,M.Phil., Ph.D;
Shankar Sriram V S, Ph.D

Abstract: Realization of the importance for advanced tool and techniques to secure the network infrastructure from the security risks has led to the development of many machine learning based intrusion detection techniques. However, the benefits and limitations of these techniques make the development of an efficient Intrusion Detection System (IDS), an open challenge. This paper presents an adaptive, and robust intrusion detection technique using Hypergraph based Genetic Algorithm (HG - GA) for parameter setting and feature selection in Support Vector Machine (SVM). Hyper - clique property of Hypergraph was exploited for the generation of initial population to fasten up the search for the optimal solution and prevent the trap at the local minima. HG-GA uses a weighted objective function to maintain the trade-off between maximizing the detection rate and minimizing the false alarm rate, along with the number of features. The performance of HG-GA SVM was evaluated using NSL-KDD intrusion dataset under two scenarios (i) All features and (ii) Features obtained from HG - GA. Experimental results show the prominence of HG-GA SVM over the existing techniques with respect to the classifier accuracy, detection rate, false alarm rate, and runtime analysis.

28 – 12 - 2016

From

V. S. Shankar Sriram Ph.D,
Associate Professor,
School of Computing,
SASTRA University,
Thanjavur – 613 401,
Tamil Nadu, India.

To

The Editor,
Knowledge – Based Systems.

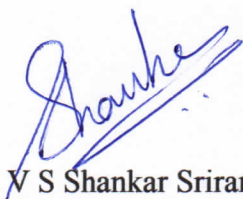
Dear Sir,

Sub. : Submission of the manuscript for publication – Reg.

I herewith submit the manuscript titled ***“An Efficient Intrusion Detection System based on Hypergraph - Genetic Algorithm for Parameter Optimization and Feature Selection in Support Vector Machine”*** for your kind consideration and publication in your esteemed Journal. I also confirm the submission is original and is not being submitted for publication elsewhere.

Thanking you

Yours Sincerely,



V S Shankar Sriram Ph.D

Title Page

Authors:

Gauthama Raman M R¹, Nivethitha Somu¹, Kannan Krithivasan², Shankar Sriram V S^{1*}

Email id:

Gauthamaraman_mr@sastra.ac.in¹, nivethitha@sastra.ac.in¹, kkannan@maths.sastra.edu², sriram@it.sastra.edu¹

Manuscript Title:

A Rough Set based Hypergraph Trust Measure Parameter Selection Technique for Cloud Service Selection

Affiliation:

¹ *Centre for Information Super Highway (CISH), School of Computing, SASTRA University, Thanjavur, Tamil Nadu, India*

² *Discrete Mathematics Research Laboratory (DMRL), Department of Mathematics, SASTRA University, Thanjavur, Tamil Nadu, India*

Corresponding Author:

Shankar Sriram V S

Email id: sriram@it.sastra.edu

Telephone: +91 4362 264101 (Extn: 2323)

Fax: +91 4362 264120

Acknowledgements

The first author thanks the Tata Consultancy Services for their financial support. The second and fourth author thanks the Department of Science and Technology, India for INSPIRE Fellowship (Grant No: DST/INSPIREFellowship/2013/963) and Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions (SR/FST/ETI-349/2013) for their financial support. The third author thanks the Department of Science and Technology—Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions Government of India (SR/FST/MSI-107/2015) for their financial support.

An Efficient Intrusion Detection System based on Hypergraph - Genetic Algorithm for Parameter Optimization and Feature Selection in Support Vector Machine

Gauthama Raman M R¹, Nivethitha Somu¹, Kannan K², Shankar Sriram V S^{1*}

¹*Centre for Information Super Highway (CISH), School of Computing, SASTRA University, Thanjavur, Tamil Nadu, India*

²*Discrete Mathematics Research Laboratory (DMRL), Department of Mathematics, SASTRA University, Thanjavur, Tamil Nadu, India*

Email id: gauthamaraman_mr@sastra.ac.in¹, nivethitha@sastra.ac.in¹, kkannan@maths.sastra.edu², sriram@it.sastra.edu^{1*}

Abstract

Realization of the importance for advanced tool and techniques to secure the network infrastructure from the security risks has led to the development of many machine learning based intrusion detection techniques. However, the benefits and limitations of these techniques make the development of an efficient Intrusion Detection System (IDS), an open challenge. This paper presents an adaptive, and robust intrusion detection technique using Hypergraph based Genetic Algorithm (HG - GA) for parameter setting and feature selection in Support Vector Machine (SVM). Hyper – clique property of Hypergraph was exploited for the generation of initial population to fasten up the search for the optimal solution and prevent the trap at the local minima. HG-GA uses a weighted objective function to maintain the trade-off between maximizing the detection rate and minimizing the false alarm rate, along with the number of features. The performance of HG-GA SVM was evaluated using NSL-KDD intrusion dataset under two scenarios (i) All features and (ii) Features obtained from HG – GA. Experimental results show the prominence of HG-GA SVM over the existing techniques with respect to the classifier accuracy, detection rate, false alarm rate, and runtime analysis.

Keywords: Hypergraph; Genetic algorithm; Support vector machine; feature subset; kernel parameters; intrusion detection systems.

Highlights

1. This paper presents a novel technique based on Hypergraph and Genetic Algorithm (HG - GA) for parameter setting and optimal feature subset selection in Support Vector Machine (SVM)
2. HG - GA exploits the hyper clique property of hypergraph for better convergence towards the optimal solution.
3. The performance of HG – GA SVM was assessed using NSL – KDD intrusion dataset under two scenarios (i) All features and (ii) Optimal feature subset obtained from HG – GA
4. The validation of HG – GA SVM was carried out with respect to the classifier accuracy, detection rate, false alarm rate, and runtime analysis.

1. Introduction

The impact of computer networks and the internet in day – to – day activities of human lives thrive the development and usage of internet based applications. It provides a global platform for the organizations and users to exchange and store sensitive information. Hence, cyber security has been a significant area of research, as it has a huge impact on each entity in the network community [1]. Several security mechanisms such as antivirus, firewall, user authentication and access control have been designed to protect the computer networks from the abnormal activities and potential attacks imposed by the intruders. However, the existing defense mechanisms have failed to safeguard the network infrastructures from the cyber – threats due to the increase in its frequency and intensity [2]. This problem can be evident from the report “The Heritage Foundation,” which lists out the security breaches on various US companies during the year 2014 [3]. Furthermore, other incidents like cyber-attacks on Estonia by Russia in 2007 [4], Russo-Georgian cyber war [5], etc. portrays the severity of cyber – threats. An intrusion can be any attempt to compromise the Confidentiality, Integrity, and Availability (CIA) or penetrate the security mechanism of the computer networks. According to NIST “*Intrusion detection* is the process of monitoring the events occurring in

a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. An *Intrusion Detection System (IDS)* is software that automates the intrusion detection process.” [6].

The development of a robust and efficient intrusion detection system remains an ongoing research problem as it deals with the dynamic environment, High Dimensional and High Sample Size (HDHSS) traffic dataset [7]. In general, intrusion detection can be viewed as a classification problem that discriminates between the normal and malicious traffic patterns [8]. Hence, for the development of a robust IDS, various machine learning and intelligent computational techniques like Artificial Neural Networks (ANN), decision trees, fuzzy logic, Principle Component Analysis (PCA), Bayesian networks, K - Nearest Neighbor (KNN), etc. were extensively used [1]. Among these techniques, Support Vector Machine (SVM) was found to be a promising tool in the design of IDS as it exhibits good performance with respect to efficiency and robustness due to structural risk minimization, high generalization ability, etc. [1]. However, a standard SVM suffers from performance - centric limitations due to feature subset selection, parameter optimization (C - penalty parameter and γ - kernel bandwidth) and imbalanced dataset [9]. To overcome these limitations, several research works were carried out based on the hybridization of SVM with various meta - heuristic techniques. However, these heuristic functions do not guarantee the global optimal solution with a better convergence rate (Table 1) [10].

Table 1 – Related Works

Authors	Technique	Application
Xu-sheng Gan et al. [11]*	Partial least square	Intrusion detection system
Hongze Li et al. [12]¥	Fruit fly optimization	Electric load forecasting
Wenchuan Wang et al. [13]¥	Mutation fruit fly optimization algorithm	Chemometrics
Liming Shen et al. [10]¥	Fruit fly optimization technique	Medical applications
Frauke Friedrichs [14]¥	Covariance matrix adaptation evolution strategy	Generic application
Shih-Wei Lin et al. [9]#	Particle swarm optimization	Generic application
Cheng-Lung Huang et al. [15]#	Continuous and discrete valued particle swarm optimization	Load balancing in distributed systems
Hui ling Chen et al. [16]#	Parallel time variant particle swarm optimization	Generic application
Seyed Mojtaba Hosseini Bamakan et al. [17]#	Time-varying chaos particle swarm optimization	Intrusion detection system
Cheng-Lung Huang [18] #	Genetic algorithm.	Generic application
Fangjun Kuang et al. [8]#	✓ Kernel principal component analysis – feature selection	Intrusion detection system
	✓ Genetic algorithm – parameter optimization	
	✓ Discrete - valued Gravitational Search Algorithm (GSA) –	
Soroor Sarafrazi et al. [19]#	Identification of the optimal feature subset	Generic application
	✓ Continuous valued GSA - Selection of kernel parameters.	
Kuan Cheng Lin et al. [20]#	Modified cat swarm optimization	Generic application
Mingyuan Zhao et al. [21]#	genetic algorithm	Generic application
Zhi Chen et al. [22] #	Coarse - grained parallel genetic algorithm	Generic application

* - Feature Selection; ¥ - Parameter Optimization; # - Feature Selection and Parameter Optimization

To address the above said challenges and to achieve the major motive behind the intrusion detection system, this work put forth a novel hypergraph - based genetic algorithm (HG - GA) technique for its application towards the development of an efficient IDS. Major contributions of this paper are as follows:

1. The representation of initial population with the application of hyper clique property of Hypergraph enhances the performance of the genetic algorithm through minimal time complexity and prevents pre - mature convergence.
2. HG - GA enhances the performance of the SVM classification model through the identification of the optimal feature subset and kernel parameters(C, γ).
3. UNB ISCX NSL - KDD, a benchmark intrusion dataset was used for experimentation and validation purposes. The performance of HG – GA SVM was evaluated with respect to the various performance metrics such as optimal feature subset, classifier accuracy, detection rate, runtime analysis, and false alarm rate.

The rest of the paper is structures as follows: Section 2 provides an insight into support vector machine, hypergraph, and genetic algorithm. Section 3 introduces the proposed technique - Hypergraph and Genetic Algorithm (HG – GA) for optimal parameter and feature subset selection. Section 4 presents the experimental settings and performance analysis of HG – GA. Section 5 concludes the paper with future works.

2. Materials and Methods

2.1 Support Vector Machine (SVM)

SVM is a kernel based supervised machine learning technique developed to address various classification and regression problems. It relies on the structural risk minimization principle, which makes them outperform the existing neural network models in a wide range of applications across various research domains like pattern recognition [23], classification [24], image processing [25], remote sensing [26], etc. According to Cristianini and Shawe-Taylor “SVMs are learning systems that use a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory” [27]. In SVM, each data sample is viewed as a point in the n-dimensional space and constructs an optimal hyperplane (decisional boundary) that separate samples of different classes (Figure 1). In the case of non - linear separable samples, non - linear kernel functions were used for the construction of hyperplane [27].

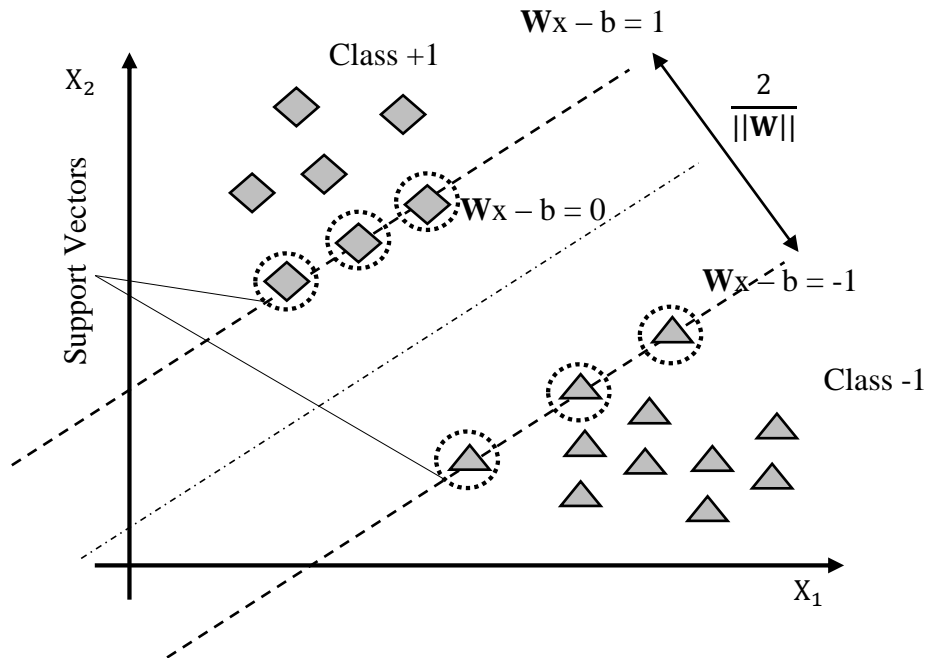


Figure 1: Linear SVM

2.1.1 Linear SVM Classifier

Consider a two-class classification problem, in which linearly separable training vectors are represented as $\{x_i, y_i\}, i = 1, 2, \dots, n$ where $x_i \in \mathbb{R}^n, y_i \in \{+1, -1\}$. The hyperplane that separates two different classes in the training vectors are [27],

$$(W \cdot x_i) + b \geq +1 \text{ for } y_i = +1 \quad (1)$$

$$(W \cdot x_i) + b \leq -1 \text{ for } y_i = -1 \quad (2)$$

Combining (1) and (2),

$$y_i[(W \cdot x_i) + b] \geq 1 \text{ for } i = 1, 2, \dots, n \quad (3)$$

Where, W is the n -dimensional coefficient weight vector normal to the hyperplane and b is the bias. Among all possible hyperplanes, SVM attempts to identify the optimal one by minimizing the Eqn.(3)

$$J(W) = \frac{1}{2} \|W\|^2 \quad (4)$$

Under the constraint $y_i(W \cdot x_i + b) \geq 1; i = 1, 2, \dots, n$. This quadric optimization problem can be solved through Lagrange function i.e. finding the saddle points of Lagrange multipliers ($a_i \geq 0$),

$$L(w, a, b) = \frac{1}{2} W^T \cdot W - \sum_{i=1}^n (a_i y_i (W \cdot x_i) + b) - 1 \quad (5)$$

For the optimal value of a_i , Eqn. (5) has to be minimized w.r.t W, b and simultaneously maximized w.r.t non-negative dual variable a_i . By differentiating L w.r.t to W and b ,

$$\frac{\partial}{\partial W} L = 0, W = \sum_{i=1}^n a_i y_i x_i \quad (6)$$

$$\frac{\partial}{\partial b} L = 0, \sum_{i=1}^n a_i y_i = 0 \quad (7)$$

From Eqn. (6) and (7), it is obvious that there exist only one minimal solution. Hence, it can be proven that there is no problem of local minima in SVM when compared with back propagation neural networks. Similarly, for the maximization of Eqn. (5), Karush Kuhn-Tucket (KKT) conditions were found to be sufficient.

Substitution of Eqn. (6) and (7) in Eqn. (5), results in a dual Lagrangian subjected to maximization w.r.t to $a_i \forall i$,

$$Max L_D(a) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j (x_i \cdot x_j) \quad (8)$$

Under the constraint,

$$\sum_{i=1}^n a_i y_i = 0; a_i \geq 0, i = 1, 2, \dots, n \quad (9)$$

Eqn.(8) is a standard quadratic optimization problem, where the solution of a_i determines the optimal value of W^*, b^* for the construction of hyperplane. Thus, the expression for the optimal hyperplane is,

$$f(x, b^*, a^*) = \sum_{i=1}^m y_i a_i^* (x_i, x) + b^* \quad (10)$$

Geometrically, the training vectors (x_i) that consist of non-zero a_i are called as support vectors, where the optimal hyperplane ($f(x, b^*, a^*)$) has the higher dependency with them. Thus, the calculation of $f(x, b^*, a^*)$ for the points lying outside the support vectors is not necessary.

2.1.2 Non-Linear SVM Classifier

For non-linear separable training vectors, the same concept can be extended by modifying Eqn. (1) and (2) as [27],

$$(\mathbf{W} \cdot x_i) + b \geq +1 - \xi_i \text{ for } y_i = +1 \quad (11)$$

$$(\mathbf{W} \cdot x_i) + b \leq -1 + \xi_i \text{ for } y_i = -1 \quad (12)$$

Where, ξ_i is the collection of slack variables, such that $\xi_i \geq 0$ for $i = 1, 2, \dots, n$. It maintains the constraint violation as small as possible, thereby minimizes the training error. Hence, Eqn. (4) can be modified as,

$$J(\mathbf{W}, \xi) = \frac{1}{2} \|\mathbf{W}\|^2 + C \sum_{i=1}^n \xi_i \quad (13)$$

subjected to $y_i(\mathbf{W} \cdot x_i + b) + \xi_i - 1; i = 1, 2, \dots, n$.

Where, C is a non-negative, user specified parameter. Eqn.(13) can be solved through maximizing the Lagrangian function given in Eqn.(8). In the case of non - linear SVM, the training vectors were transformed into a higher dimensional space with the use of a mapping function (Kernel function) [27]. Hence, Eqn. (8) is altered as,

$$\text{Max } L_D(a) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j < \phi(x_i), \phi(x) > \quad (14)$$

Under the constraint,

$$\sum_{i=1}^n a_i y_i = 0; 0 \leq a_i \leq C, i = 1, 2, \dots, n \quad (15)$$

Where, $< \phi(x_i), \phi(x) >$ is the kernel function. Various kernel functions used in non - linear SVM are discussed in Table 2 [27]. Thus, the expression for the optimal hyperplane is,

$$f(x, b^*, a^*) = \sum_{i=1}^m y_i a_i^* < \phi(x_i), \phi(x) > + b^* \quad (16)$$

Table 2: Kernel Functions

Kernel	Function
Linear	$K(x_i, x) = (x_i \cdot x)$
Polynomial Function	$K(x_i, x) = ((x_i \cdot x) + 1)^d$
Radial Basis Function	$K(x_i, x) = \exp(-\gamma x_i - x ^2)$
Bspline kernel Function	$K(x_i, x) = d(x_i - x)$
Sigmoid Function	$K(x_i, x) = \tanh(d1(x_i \cdot x) + d2)$
Fourier Kernel Function	$K(x_i, x) = (\sin\left(d + \frac{1}{2}\right)(x_i - x)) / (\sin(\frac{1}{2}(x_i - x))$

2.2 Genetic Algorithm (GA)

Genetic algorithm - inspired by the Darwinian principle of “Survival of Fittest” is an adaptive, meta - heuristic-based optimization technique [28]. It is based on the approach that “all individuals in a generation will compete with each other for resources and the most successful individuals were only allowed to produce their offspring.” In general, GA is a population-based search methodology, in which each candidate is represented as a chromosome of fixed length binary string. Apart from this, chromosomes can also be represented as value encoding, tree encoding permutation encoding, etc. The quality of each chromosome is assessed using the fitness function for the selection of the fittest one to represent the offsprings of the next generation. The exploration and exploitation mechanism in larger search space proves the efficiency of the genetic algorithm hence, it guarantees the global optimal solution. Algorithm 1 describes the generic structure of GA. The important operators used in the generation of offspring are discussed below [28].

(i) Selection

During the selection process, genes of the chromosomes in the current generation are transferred to the next generation based on its fitness evaluation value. The selection probability of the chromosome is high for the fittest chromosome (high fitness value). Some of the selection techniques in GA are roulette wheel selection, linear ranking selection, exponential ranking selection, tournament selection, Boltzmann selection, etc.

(ii) Crossover

In the crossover, a random mechanism is employed to interchange the gene information between two parent chromosomes. With the use of crossover points, each parent chromosome is divided into two namely head and tail. New offspring is obtained by linking head of first parent chromosome with the tail of the second parent chromosome and vice versa as shown in Figure 2. Various crossover techniques used in GA are single point crossover, two crossover, uniform crossover, etc.

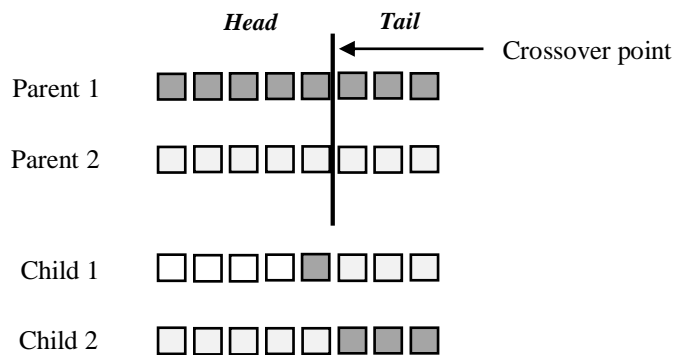


Figure 2: Crossover Operation

(iii) Mutation

Mutation is a process, which is carried out after the crossover operation for introducing randomness into the solution. In this process, the positions for the mutation were chosen in a random fashion, and the bits in the respective positions were flipped as shown in Figure 3. For example, in a binary representation, the parent 10101110 is mutated to produce child 10011110. Similarly, for other representations various mutation techniques like uniform and non-uniform technique, boundary method, Gaussian method, etc. can be used.

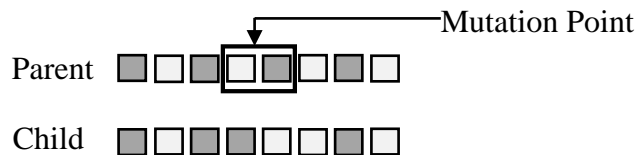


Figure 3: Mutation Operation

Algorithm 1: Genetic Algorithm

Input:

GA Parameters

Output:

Best Solution

GA()

1. **begin**
 2. $t \leftarrow 0$
 3. **Generate population** in a random fashion ($Pop(t)$)
 4. **Calculate the fitness value** of $Pop(t)$
 5. **while** (!Termination condition) **begin**

 // Generation of offsprings

6. **Select the best individuals** among $Pop(t)$
 7. Apply **crossover operation**
 8. Apply **mutation operation**
 9. Evaluate **the fitness value**
 10. $t \leftarrow t + 1$
 11. **end**
 12. **return** best solution
-

3.1 Hypergraph

A Hypergraph is a mathematical framework and generalization of traditional graph theory in which each (hyper) edge covers one or more vertices to represent the multiple relationships (n-ary relations) among them. As hypergraph can express the topological and geometrical relationship among the variables in a significant manner, computational models developed based on the properties of hypergraph possess minimal complexity. The efficiency and applicability of the hypergraph were proven in many real-time applications like image processing [29], network security [30], cloud computing [31], medical diagnosis [32], etc. This section provides an insight to some basic definitions of the hypergraph and its clique property, which can be hybridized with GA to enhance the performance of support vector machine.

Definition 1:(*Hypergraph*) A hypergraph H (Figure 4) is formulated as a couple (V', \mathcal{E}') , where V' is the non-empty finite set called vertices and $\mathcal{E}' = (e_i)_{i \in J}$ is the non-empty subsets of V called as hyperedges, such that $\cup e_i = V, i \in J$ where $J = \{1, 2, \dots, n\}, n \in \mathbb{N}$ [33].

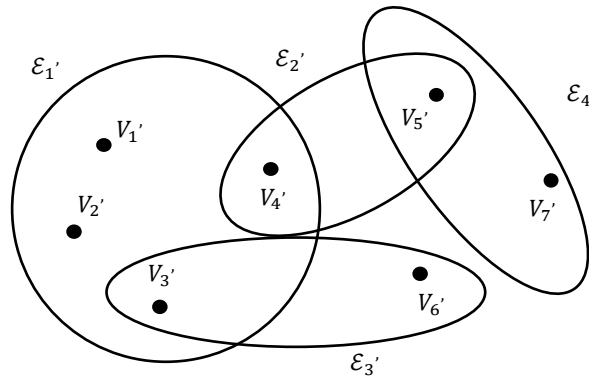


Figure 4: Hypergraph $H = \{V', \mathcal{E}'\}; V' = \{V_1', V_2', V_3', V_4', V_5', V_6', V_7'\}; \mathcal{E}' = \{\mathcal{E}_1', \mathcal{E}_2', \mathcal{E}_3', \mathcal{E}_4'\}; \mathcal{E}_1' = \{V_1', V_2', V_3', V_4'\}; \mathcal{E}_2' = \{V_4', V_5'\}; \mathcal{E}_3' = \{V_3', V_6'\}; \mathcal{E}_4' = \{V_5', V_7'\}$

Definition 2: (*Representative Graph*) Given a hypergraph $H = \{\mathcal{E}_1', \mathcal{E}_2', \dots, \mathcal{E}_n'\}$ on V' then the line graph or the representative graph of H is a graph (V, E) such that [33]

1. $V = JorV = \mathcal{E}$ when H has no repeated hyperedges
2. $\{l, m\} \in E \mid l \neq m$ if and only if $e_l \cap e_m \neq \emptyset$

Definition 3: (*Closed Neighborhood*) Given a graph $G = \{V, E\}$, let $q \in V$, then the closed neighborhood of q in G can be defined as [29,33]

$$\theta(q) = \{p \in V \mid p \text{ is adjacent to } q \text{ in } G\} \cup \{q\} \quad (17)$$

Definitions 4: (*Complete Graph*) Given an undirected graph $G = \{V, E\}$, which consists of m vertices ($V = \{V_1, V_2, \dots, V_m\}$) and n edges ($E = \{e_1, e_2, \dots, e_n\}$) is said to be *complete* (Figure 5), if there exists an edge $e \in E$ between the vertices $(V_i, V_j) \in V, \forall i, j = \{1, 2, \dots, m\} \mid i \neq j$. [33]

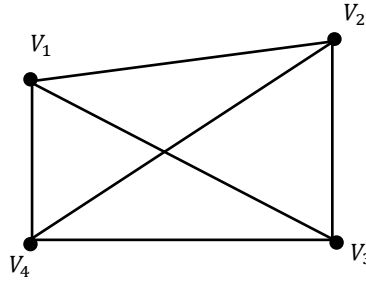


Figure 5: Complete graph

Definition 5: (*K-Clique*) For a given undirected graph $G = \{V, E\}$, K -clique of G (Figure 6) is a subset $C \subset V$ & $K = |C|$, such that subgraph C satisfy the condition of a complete graph [33]

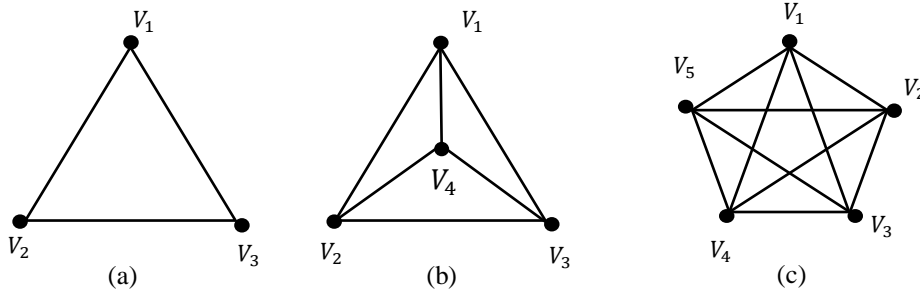


Figure 6: (a) 3-Clique (b) 4-Clique (c) 5-Clique

Proposition 5.1: Consider a graph $G = \{V, E\}$ and $C = \{V_1, V_2, \dots, V_r\}$, then C is the clique of G , if and only if $C \subset \bigcap_{i=1}^r \theta(V_i)$ (Figure 7)

Proof:

Consider C be a clique of G . Let $q \in V$, then $q \in \theta(V_j), \forall j = \{1, 2, \dots, r\}$ and $C \subset \bigcap_{i=1}^r \theta(V_i)$. Conversely, consider $C \subset \bigcap_{i=1}^r \theta(V_i)$. Let l, n be a distinct element of C , then by hypothesis $q \in \theta(V_j), \forall j = \{1, 2, \dots, r\}$, l , and n are adjacent. Hence, C is a clique of G .

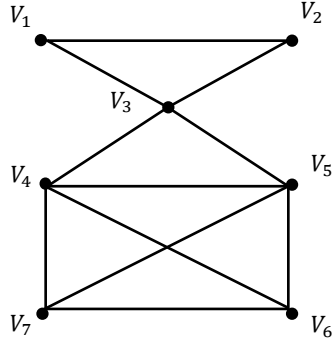


Figure 7: A Simple Graph G with 6 Vertices and 11 Edges; All possible cliques of G is $C_1 = \{V_1, V_2, V_3\}, C_2 = \{V_3, V_4, V_5\}, C_3 = \{V_4, V_5, V_6, V_7\}, \dots, C_n$.

4. HG GA – SVM: The Proposed Methodology

This section discusses the logic behind the proposed Hypergraph based Genetic Algorithm (HG – GA) technique for parameter setting and feature subset selection in SVM (Algorithm 2). The working of HG – GA comprises of two predominant sections namely (i) Optimal feature subset selection and parameter optimization (ii) Estimation of SVM performance (Figure 8). In the former, both kernel parameters and feature subset for SVM were dynamically adjusted through the fitness function (HG – GA), while in the latter, the optimal kernel parameters and feature subset obtained from HG - GA were fed to SVM for classification. The workflow of HG – GA can be described using a sample dataset (Table 3) as follows:

Table 3: Sample Dataset

Sample	Features					Decision Class
	F_1	F_2	F_3	F_4	F_5	
S_1	u_{11}	u_{21}	u_{31}	u_{41}	u_{51}	D_1
S_2	u_{12}	u_{22}	u_{32}	u_{42}	u_{52}	D_2
S_3	u_{13}	u_{23}	u_{33}	u_{43}	u_{53}	D_3
S_4	u_{14}	u_{24}	u_{34}	u_{44}	u_{54}	D_4
S_5	u_{15}	u_{25}	u_{35}	u_{45}	u_{55}	D_5
S_6	u_{16}	u_{26}	u_{36}	u_{46}	u_{56}	D_6
S_7	u_{17}	u_{27}	u_{37}	u_{47}	u_{57}	D_7
S_8	u_{18}	u_{28}	u_{38}	u_{48}	u_{58}	D_8
S_9	u_{19}	u_{29}	u_{39}	u_{49}	u_{59}	D_9
S_{10}	u_{110}	u_{210}	u_{310}	u_{410}	u_{510}	D_{10}

(i) Generation of Training and Testing Dataset

From the given input dataset S of N samples, the training and testing dataset ($Training_{Data}$ and $Testing_{Data}$) were generated in the ratio of 80:20. With respect to Table 3, S_1 to S_8 belongs to $Training_{Data}$ and S_9 & S_{10} belongs to $Testing_{Data}$.

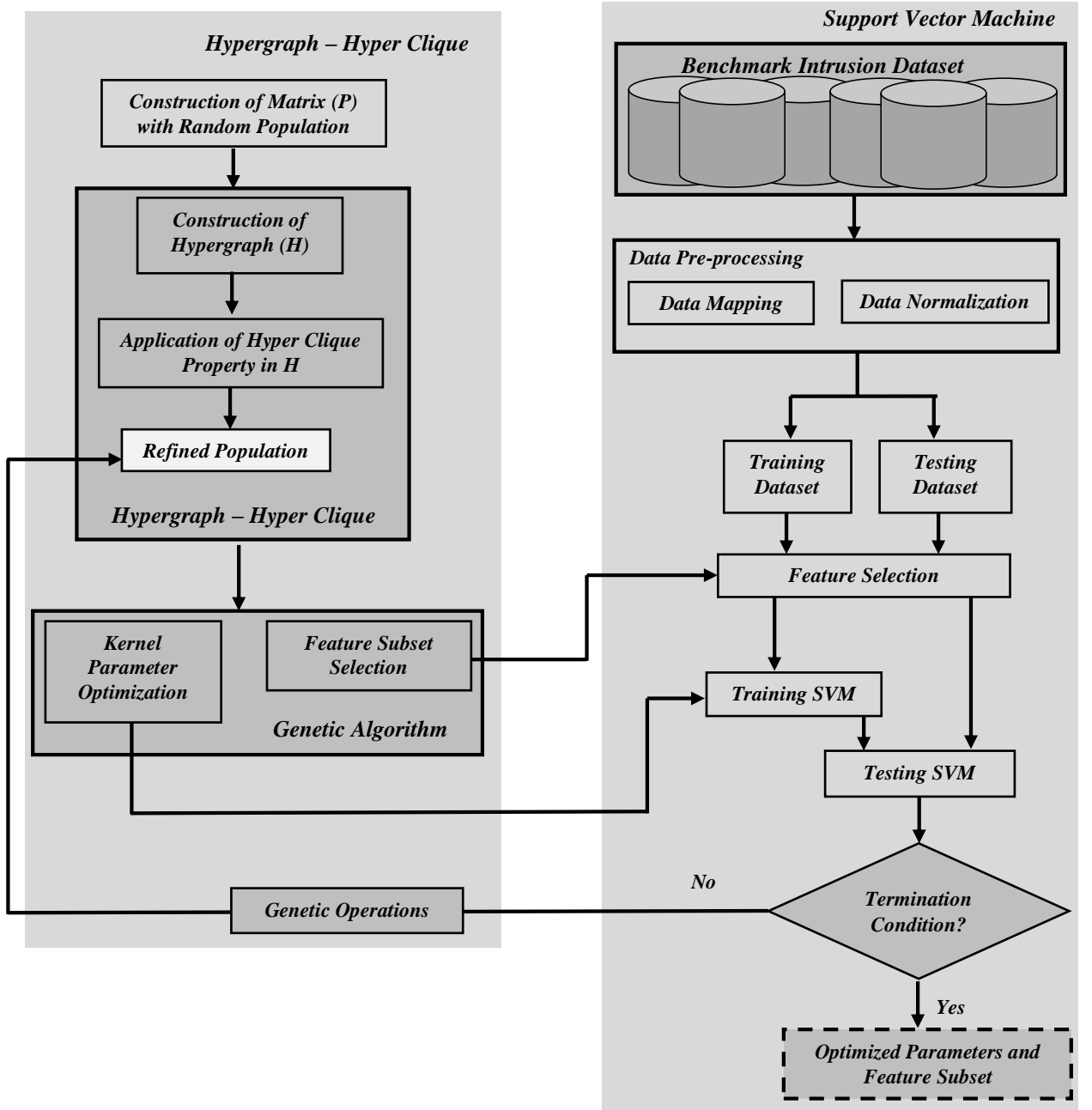


Figure 8: Hypergraph and Genetic Algorithm (HG – GA) based Technique for Parameter Setting and Feature Selection in Support Vector Machine

(ii) **Generation of Initial Chromosomes and Selection of Kernel Parameters and Feature Subset**

In GA, each individual in the population was generated in a random manner and the fittest individual identified using the fitness function undergoes crossover and mutation operations. Similarly, in HG – GA, the generation of initial population involves three phases: (i) Construction of a random two-dimensional matrix (P) of order $M \times K$, where M and K represents the total number of individuals in the population and size of each individual respectively; (ii) Construction of hypergraph structure (H) from P ; (iii) Application of hyper clique property on the hypergraph (H) and generation of initial population (Figure 9). The individuals in the population were denoted as the binary vector. Each individual in the population comprises of three major parts, $([P_{i,1} - P_{i,c_n}])$, $([P_{i,1} - P_{i,\gamma_n}])$ and $(P_{i,1} - P_{i,f_n})$, which

represents the value of the kernel parameters($C \& \gamma$) and feature subset respectively as in Figure 9. C_n and γ_n are the number of bits used to represent $C \& \gamma$ values respectively, which are chosen based on the lower and upper bounds kernel parameter values. Similarly, f_n is the number of bits used to represent the feature subset, whose value is same as the number of features in the input dataset [18].

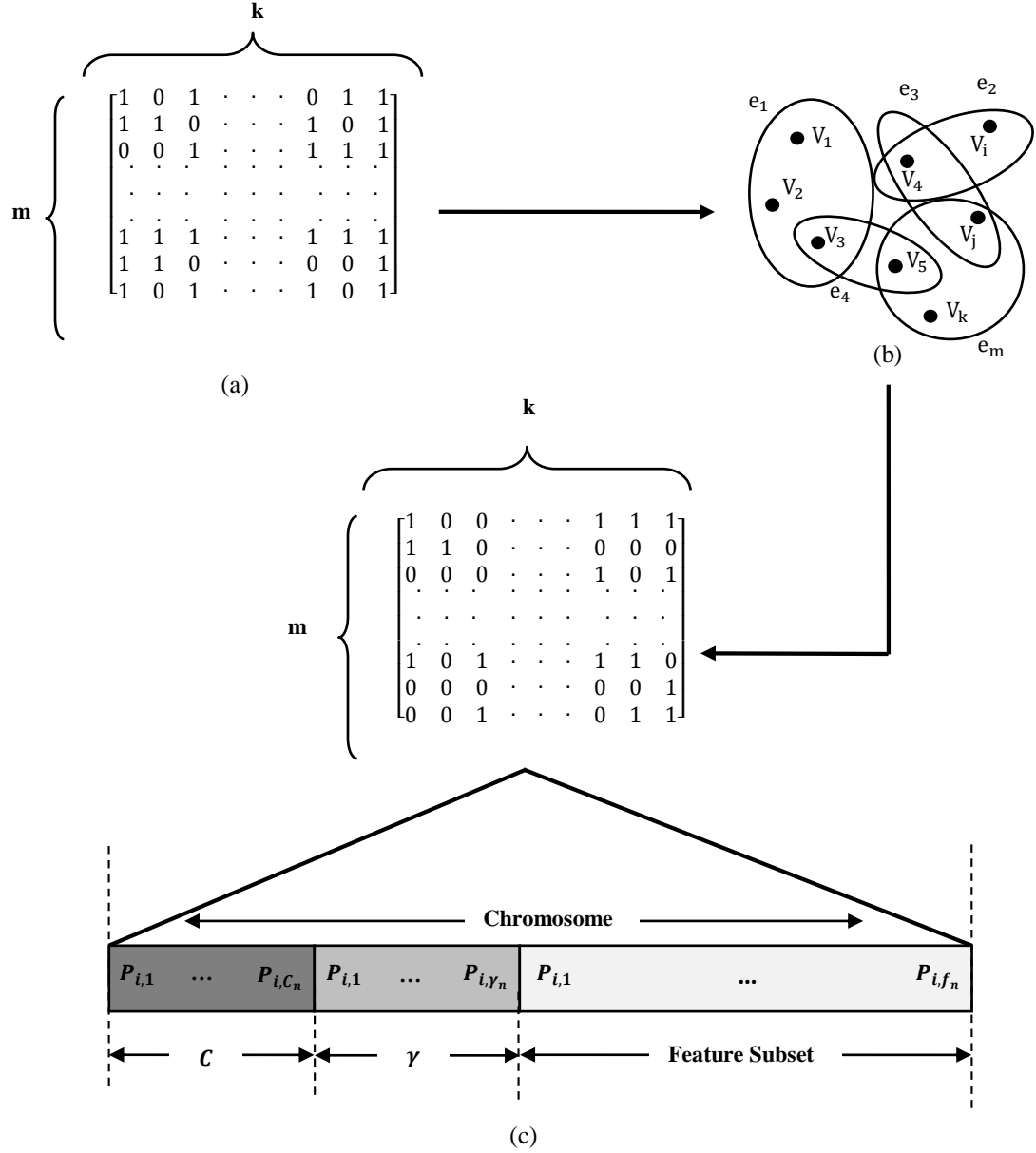


Figure 9: Generation of Initial Population; (a) Random Binary Strings; (b) Construction of Hypergraph and Application of Hyper Clique Property; (c) Initial Population

After the construction of the initial population, the binary vectors $([P_{i,1} - P_{i,c_n}], [P_{i,1} - P_{i,\gamma_n}])$ were converted into their equivalent floating point representation using Eqn. (18) [18].

$$V = Min_V + \frac{Max_V - Min_V}{2^{L-1}} x d \quad (18)$$

where, V is the floating point value of the parameter, Max_V & Min_V is the maximum and minimum value of the parameter, L is the length of the binary vector, and d is the decimal value of the binary vector.

Similarly, the binary vector that represent the feature subset was processed for further evaluation i.e. '1' – presence of the feature, '0' – neglected feature. Based on the feature subset in each chromosome, unrelated features from $Training_{Data}$ and $Testing_{Data}$ were neglected from the further process.

Algorithm 2: HG GA Technique for Parameter Setting and Feature Selection

Input

//Input Dataset

$S \leftarrow \{(x_1, y_1), (x_2, y_2) \dots, (x_N, y_N)\};$

$x_i \leftarrow$ Conditional feature vector

$y_i \leftarrow$ Decisional attribute

//Genetic Algorithm Parameters

$Gen_{Max} \leftarrow$ Maximum Generation

$M \leftarrow$ Number of Population

//Fitness Function Parameters

$D_W \leftarrow$ Weight for Detection rate

$FA_W \leftarrow$ Weight for False Alarm Rate

$F_W \leftarrow$ Weight for feature subset size

Output

$C^*, \gamma^* \leftarrow$ Optimal SVM Parameters

$F^* \leftarrow$ Optimal feature subset

HG GA-SVM ()

1. **Begin**
 2. Initialize the value of $t \leftarrow 1$, mutation rate and crossover rate
 3. Generate training dataset ($Training_{Data}$), testing dataset ($Testing_{Data}$) from S
 4. *//Generation of Initial Population for Genetic Algorithm*
 4. Construct a matrix $[R]_{M \times K}$ which consists of randomly generated binary strings
 5. Build a hypergraph H using $[R]_{M \times K}$
 6. Generate the initial population $[P]_{M \times K}^t$ using the Hyper clique property of hypergraph (Proposition 5.1)
 7. **while** (! Termination Condition) **begin**
 8. **for each** $i \leftarrow 1$ to Pop_{Size} **begin**
 9. Extract the value of the parameters (C, γ) and feature subset from i^{th} chromosome
 10. Compute fitness value of the i^{th} chromosome (Eqn.(19))
 11. **end for**
 12. Parents Selection using Tournament selection
 13. Apply Crossover & Mutation
 14. Population Updation
 15. $t++$
 16. **end while**
 17. **return** C^*, γ^*, f^*
 18. **end**
-

(iii) **Train SVM**

Train the SVM using $Training_{Data}$ with the kernel parameters and feature subset obtained from each chromosome (Step ii). For example, SVM is trained with the kernel parameters & feature subset generated from the population P_1, P_2, P_3 and P_4 as demonstrated in Table 4 along with the $Training_{Data}$ ($S_1 - S_8$)

Table 4: Generation of Kernel Parameters and Feature Subset from Chromosomes

Population	Chromosome										C	γ	Feature subset
P_1	1	...	0	1	...	1	1	0	1	0	0.73	0.24	$\{f_1, f_2, f_3, f_5\}$
P_2	0	...	1	1	...	0	1	0	0	1	0.22	0.44	$\{f_1, f_5\}$
P_3	0	...	0	0	...	1	1	1	1	1	0.61	0.52	$\{f_1, f_2, f_3, f_4, f_5\}$
P_4	1	...	1	1	...	0	1	0	0	1	0.13	0.76	$\{f_2, f_5\}$

(iv) **Test SVM**

Evaluates the performance of SVM using $Testing_{Data}$ based on the fitness function (number of features, detection rate, and false alarm rate) given in Eqn. (19). Each sample in the $Testing_{Data}$ is processed with respect to the feature subset obtained from the population P_i and tested with the trained SVM (Step iii).

(v) **Fitness Evaluation**

In this work, in addition to the detection rate and false alarm rate, number of features was considered as a predominant parameter to evaluate the performance of an IDS, as it has a huge impact on the complexity of the learning model. Hence, the fitness function of HG – GA was designed based on the predominant parameters (number of features, detection rate, and false alarm rate) [17]. Detection rate and false alarm rate were estimated with the use of True Positive (T_p), True Negative (T_N), False Positive (F_p), and False Negative (F_N) as shown in Table 5.

True Positive (T_p): Measure of *malicious behaviors* correctly identified as an *attack*.

True Negative (T_N): Measure of *benign behaviors* correctly identified as *benign*.

False Positive (F_p): Measure of *benign behaviors* misclassified as an *attack*.

False Negative (F_N): Measure of *malicious behaviors* misclassified as *benign*.

Table 5: Estimation of Detection Rate and False Alarm Rate

Performance Metric	Equation
Detection rate (DR)	$\frac{T_p}{T_p + F_N}$
False Alarm Rate (FAR)	$\frac{F_p}{F_p + T_N}$

To achieve the optimal fitness value, the selected chromosome has to maximize the detection rate and minimize the false alarm rate & number of features. This type of Multi - Criteria Decision Making (MCDM) problem can be solved by a single weighted fitness function, which combines three independent tasks into a single goal. As in Eqn. (19), three predefined weights D_W , FA_W and F_W were associated with the detection rate, false alarm rate and number of features respectively [17]. To achieve the goal (optimal fitness value), D_W was set with a higher value among the rest, since the detection rate has to be maximized, whereas the false alarm rate and number of features have to be minimized.

$$Fitness_{HG-GASVM} = D_W[DR] + FA_W[1 - FAR] + F_W \left[1 - \frac{\sum_{i=1}^N F_i}{N} \right] \quad (19)$$

where,

$N \leftarrow$ Number of features

$$F_i \leftarrow \begin{cases} 1 & \text{corresponds to the existence of } i^{th} \text{ feature} \\ 0 & \text{corresponds to the absence of } i^{th} \text{ feature} \end{cases}$$

The ideology behind HG – GA is that the presence of sufficient number of 1's in the initial population generated by the application of hypergraphs' hyper clique property makes the fitness function to reach its optimal value in less number of iterations. Thereby, minimizing the convergence time of GA.

(vi) **Termination Condition**

Once the termination condition (maximum fitness value or generation) was achieved, the algorithm returns the optimal value of the kernel parameters (C^*, γ^*) and optimal feature subset f^* . Otherwise, goto step (vii).

(vii) **Genetic Operators**

Select the fittest chromosome using tournament selection and use crossover and mutation operators to generate offsprings for the next generation. Repeat the process from step ii.

4. Experimental Analysis and Discussions

4.1 Dataset Description

KDD CUP 1999, a benchmark and most commonly used intrusion dataset for the evaluation of the different intrusion detection techniques. It was derived from the DARPA 98 dataset (DARPA 98 IDS evaluation programme, managed by Lincoln Labs), which consists of five million connection records simulated from the U.S Airforce military environment. However, Tavallaee et al. proposed NSL-KDD dataset [34], due to various limitations in KDD cup 1999 like uneven distribution of samples, redundancy and duplication in records, etc. [35]. Each sample in the NSL-KDD dataset is described by 41 conditional attributes followed by a class label. The conditional attributes, which are of discrete and continuous in nature can be categorized into (i) Basic features (ii) Contents features (iii) Time-based traffic features (iv) Host-based traffic features (Table 6). Any network behavior that deviates from "Normal" is considered to be an attack (class label). NSL-KDD dataset addresses 24 types of attacks which can be grouped into DoS, U2R, R2L and Probe as given in Table 7.

Table 6: Attributes in NSL-KDD dataset

Class	S.No	Feature Name	Class	S.No	Feature Name
Basic Features	F1	Duration*	Same Host Features	F23	Count*
	F2	Protocol Type**		F24	Srv Count*
	F3	Service**		F25	Serror Rate*
	F4	Flag**		F26	Srvserror Rate*
	F5	Source Bytes*		F27	Rerror Rate*
	F6	Destination Bytes*		F28	Srvrerror Rate*
	F7	Land**		F29	Same Srv Rate*
	F8	Wrong Fragment*		F30	Diff Srv Rate*
	F9	Urgent*		F31	Srv Diff Host Rate*
Content Features	F10	Hot*	Same Services Features	F32	Dst Host Count*
	F11	Number Failed Logins*		F33	Dst Host Srv Count*
	F12	Logged In**		F34	Dst Host Same Srv Rate*
	F13	Num Compromised*		F35	Dst Host Diff Srv Rate*
	F14	Root Shell*		F36	Dst Host Same Src Port Rate*
	F15	Su Attempted*		F37	Dst Host Srv Diff Host Rate*
	F16	Num Root*		F38	Dst Host Serror Rate*
	F17	Num File Creations*		F39	Dst Host Srvserror Rate*

F18	Num Shells*		F40	Dst Host Rerror Rate*
F19	Num Access Files*		F41	Dst Host Srvrerror Rate*
F20	Num Outbound Cmds*		F42	Decision Label**
F21	Is Host Login**			
F22	Is Guest Login**			

* Continuous type ** Discrete type

Table 7: Attacks in NSL-KDD dataset

Attacks			
Deniel of service (DoS)	User to root (U2R)	Remote to local (R2L)	Probing (Probe)
Smurf	Buffer over flow	Ftp write	Ipsweep
Back	Load module	Guess password	Nmap
Land	Perl	Imap	Satan
Neptune	Perl	Mutlihop	Portsweep
Pod		Spy	
Tear drop		Phf	
		Warezcilent	
		Warezmaster	

4.2 Experimental Setup

The proposed technique (HG – GA) was implemented using MATLAB 6.5 with LIBSVM version 3.20 package developed by Chang and Lin [36]. Experiments were carried out in an INTEL® Core™ i5 processor @ 2.40 GHz system with 8 GB RAM running Windows 7 operating system, and WEKA tool was used for validation purposes. During the initial stage of the experiment, data preprocessing technique was carried out to transform the NSL-KDD dataset into a compatible format supported by HG – GA SVM. Data mapping technique was used to transform the discrete value of the features into numeric value as demonstrated in Table 8.

Table 8: Samples Before and After Data Mapping Technique

Samples Before Data mapping
0,tcp,http,SF,310,1512,0,0,0,0,1,0,0,0,0,0,0,0,0,13,13,0.00,0.00,0.00,0.00,1.00,0.00,0.00,246,255,1.00,0.00,0.00,0.01,0.00,0.00,0.00,0.00,normal
0,tcp,iso_tsap,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,194,9,1.00,1.00,0.00,0.00,0.05,0.08,0.00,255,9,0.04,0.09,0.00,0.00,1.00,1.00,0.00,0.00,anomaly
0,tcp,http,SF,286,366,0,0,0,0,1,0,0,0,0,0,0,0,0,0,7,7,0.00,0.00,0.00,0.00,1.00,0.00,0.00,48,255,1.00,0.00,0.02,0.04,0.00,0.00,0.00,0.00,normal
0,tcp,other,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,508,1,0.11,0.00,0.89,1.00,0.00,1.00,0.00,255,1,0.00,1.00,0.00,0.00,0.18,0.00,0.82,1.00,anomaly
0,tcp,http,SF,54540,8314,0,0,0,2,0,1,1,0,0,0,0,0,0,0,0,4,10,0.00,0.00,0.00,0.00,1.00,0.00,0.20,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.04,0.04,anomaly
0,tcp,http,SF,222,333,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,137,237,1.00,0.00,0.01,0.03,0.00,0.00,0.00,0.00,normal
0,udp,private,SF,28,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,92,92,0.00,0.00,0.00,0.00,1.00,0.00,0.00,112,92,0.82,0.03,0.82,0.00,0.00,0.00,0.00,normal
0,tcp,http,SF,303,315,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,22,22,0.00,0.00,0.00,0.00,1.00,0.00,0.00,72,255,1.00,0.00,0.01,0.02,0.00,0.00,0.00,0.00,normal
0,tcp,other,RSTR,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,1.00,1.00,1.00,0.00,0.00,255,1,0.00,0.94,0.93,0.00,0.00,0.00,0.93,1.00,normal
0,udp,domain_u,SF,42,42,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9,13,0.00,0.00,0.00,0.00,1.00,0.00,0.15,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,normal
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,101,14,0.00,0.00,1.00,1.00,0.14,0.06,0.00,255,14,0.05,0.05,0.00,0.00,0.00,0.00,1.00,1.00,anomaly

0,tcp,http,SF,346,61911,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,212,255,1.00,0.00,0.00,0.02,0.00,0.00,0.00,0.00,normal
0,tcp,finger,RSTO,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,190,2,0.00,0.00,1.00,1.00,0.01,0.07,0.00,255,2,0.01,0.07,0.00,0.00,0.00,0.00,1.00,1.00,anomaly
Samples After Data mapping
0,1,1,1,310,1512,0,0,0,0,0,1,0,0,0,0,0,0,0,0,13,13,0.00,0.00,0.00,0.00,1.00,0.00,0.00,246,255,1.00,0.00,0.00,0.01,0.00,0.00,0.00,0.00,1
0,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,194,9,1.00,1.00,0.00,0.00,0.05,0.08,0.00,255,9,0.04,0.09,0.00,0.00,1.00,1.00,0.00,0.00,2
0,1,1,1,286,366,0,0,0,0,0,1,0,0,0,0,0,0,0,0,7,7,0.00,0.00,0.00,0.00,1.00,0.00,0.00,48,255,1.00,0.00,0.02,0.04,0.00,0.00,0.00,0.00,1
0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,508,1,0.11,0.00,0.89,1.00,0.00,1.00,0.00,255,1,0.00,1.00,0.00,0.00,0.18,0.00,0.82,1.00,2
0,1,1,1,54540,8314,0,0,0,2,0,1,1,0,0,0,0,0,0,0,0,4,10,0.00,0.00,0.00,0.00,1.00,0.00,0.20,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.04,0.04,2
0,1,1,1,222,333,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,137,237,1.00,0.00,0.01,0.03,0.00,0.00,0.00,0.00,1
0,2,4,1,28,0,0,3,0,0,0,0,0,0,0,0,0,0,0,92,92,0.00,0.00,0.00,0.00,1.00,0.00,0.00,112,92,0.82,0.03,0.82,0.00,0.00,0.00,0.00,1
0,1,1,1,303,315,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,22,22,0.00,0.00,0.00,0.00,1.00,0.00,0.00,72,255,1.00,0.00,0.01,0.02,0.00,0.00,0.00,0.00,1
0,1,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,1.00,1.00,1.00,0.00,0.00,255,1,0.00,0.94,0.93,0.00,0.00,0.00,0.00,0.93,1.00,1
0,2,5,1,42,42,0,0,0,0,0,0,0,0,0,0,0,0,0,9,13,0.00,0.00,0.00,0.00,1.00,0.00,0.15,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,1
0,1,4,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,101,14,0.00,0.00,1.00,1.00,0.14,0.06,0.00,255,14,0.05,0.05,0.00,0.00,0.00,0.00,0.00,1.00,2
0,1,1,1,346,61911,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,212,255,1.00,0.00,0.00,0.02,0.00,0.00,0.00,0.00,1
0,1,6,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,190,2,0.00,0.00,1.00,1.00,0.01,0.07,0.00,255,2,0.01,0.07,0.00,0.00,0.00,0.00,0.00,1.00,2

On to the next phase, for the generation of training and testing datasets, random sampling without replacement technique was carried out on KDDTrain+ & KDDTest+ dataset [34]. Finally, to show the predominance of the HG-GA, it was compared with few existing techniques like PSO-SVM, GE-SVM, Grid-SVM, Random Forest and Bayes Net. As these techniques were available as libraries in WEKA and MATLAB, they have not been implemented separately [37]. The parameter setting for HG – GA is shown in Table 9 and for the existing techniques is same as in [10,15]. In addition, 10 fold cross-validation technique and following metrics were used to quantify the performance of the classifiers,

$$\text{Accuracy}(Acc) = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

$$\text{Detection Rate (DR)} = \frac{T_P}{T_P + F_N}$$

$$\text{False Alarm Rate (FAR)} = \frac{F_P}{F_P + T_N}$$

Table 9: Parameter values for HG-GA SVM

S.No	Parameters	Value
1	C	$[2^{-5} - 2^5]$
2	γ	$[2^{-4} - 2^4]$
3	Number of Iterations(N)	500
4	Number of Populations (Pop _{Size})	8

5	Weights	
	i. D_W	0.80
	ii. FA_W	0.05
	iii. F_W	0.15
6	Crossover rate	80%
7	Mutation rate	2%

4.3 Experimental Results and Discussions

The effectiveness of HG – GA SVM in its ability to differentiate between the normal and malicious behavior of the network traffic patterns was proven in terms of (i) Classification accuracy (ii) Detection rate (iii) False alarm rate and (iv) Runtime analysis. Major goal behind the development of an efficient and robust IDS is to maximize (i) and (ii) & to minimize (iii) & (iv). Initially, the performance of HG-GA SVM was assessed with respect to the classification accuracy i.e. number of samples which are correctly identified as "Normal" and "Attack." Table 10 represents the experimental results with respect to the classification accuracy carried out under two different scenarios (Scenario 1: All features under consideration; Scenario 2: Features in the feature subset obtained from feature selection technique). From Table 10, it is clear that under both scenarios HG-GA SVM performs well in terms of classification accuracy. In addition, the proposed technique achieves improved classification accuracy (approx. 2%), when trained with the optimal feature subset.

Table 10: Performance comparison based on Classification Accuracy

Classifier	Without Feature Selection Technique	With Feature Selection Technique	
	Accuracy (%)	Accuracy (%)	No. of features
<i>Grid-SVM</i>	89.45	91.36	32
<i>PSO-SVM</i>	92.94	93.49	36
<i>GA-SVM</i>	94.58	95.32	35
<i>Random Forest</i>	95.16	NA	NA
<i>Bayes Net</i>	94.24	NA	NA
<i>HG-GA SVM</i>	95.32	97.14	35

Subsequently, the performance of the classifiers was evaluated based on the detection rate and false alarm rate (Table 11). It was interesting to note that, (i) Scenario 1: the detection rate of HG – GA SVM was similar to that of Random Forest, however the false alarm rate of the former was found to be lower than that of the latter; (ii) Scenario 2: The detection rate was higher and false alarm rate was lower than the existing classifiers. Hence, it proves the need and role of an efficient feature selection technique in the design of a robust IDS.

Table 11: Performance Analysis based on Detection Rate and False alarm rate

Classifier	Without Feature Selection Technique		With Feature Selection Technique	
	Detection Rate	False Alarm Rate	Detection Rate	False Alarm Rate
<i>Grid-SVM</i>	90.13	5.26	92.75	2.45
<i>PSO-SVM</i>	94.54	3.92	94.34	1.09
<i>GA-SVM</i>	95.33	3.04	95.89	0.92
<i>Random Forest</i>	95.98	4.87	NA	NA
<i>Bayes Net</i>	95.33	4.12	NA	NA
<i>HG-GA SVM</i>	95.82	3.17	96.72	0.83

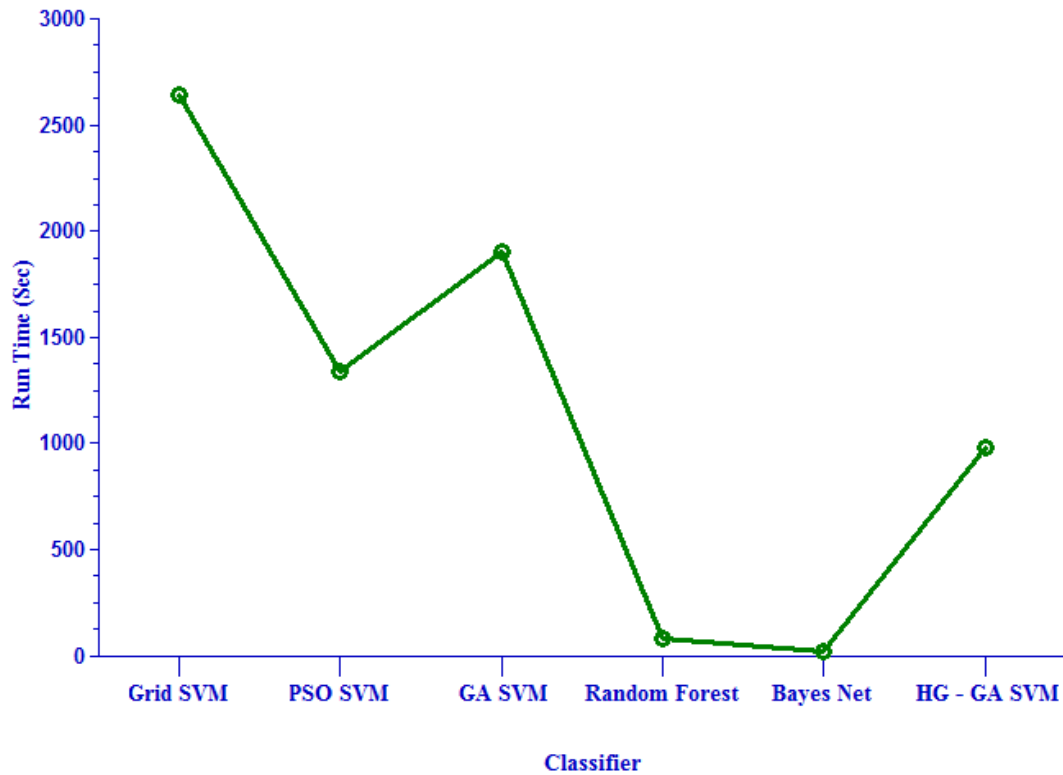


Figure 10: Performance analysis based on Runtime analysis

Finally, the performance of HG-GA SVM was carried out based on the time complexity i.e. total run time taken by the technique to perform its intended work. Figure 10 illustrates the run time of each algorithm during the process of 10 fold cross validation. The run time of Random Forest and Bayes Net was found to be minimal as it does not search for the optimal feature subset. The main reason behind the high run time of Grid-SVM, PSO-SVM and GA-SVM was due to several operations such as feature subset generation, population initialization, reproductive operators, etc. Out of the various meta - heuristic based classifiers, the run time of HG-GA SVM to identify the optimal kernel parameters and feature subset for SVM was found to be minimal.

Table 12: Performance Analysis of HG – GA SVM with the Recent Development in IDS

Authors	Detection Rate	False Alarm Rate
Kuang et al. [8]*	95.26	1.03
Singh et al. [38]**	97.67	1.74
De la Hoz et al. [39]**	93.4	14
Tavallae et al. [35]**	80.67	NA
Tsang et al. [40]*	92.76	NA
Kayacik et al. [41]*	90.6	1.57
Bamakan SM et al. [17]**	97.03	0.87
HG – GA SVM**	96.72	0.83

*KDD cup dataset **NSL-KDD cup dataset

Table 12 presents the comparative analysis of HG – GA SVM with the recent advancements in IDS based on the detection rate and false alarm rate. From Table 12, it is evident that HG – GA SVM ranks first and third with respect to detection rate and false alarm rate respectively. However, we cannot claim that HG – GA SVM outperforms the existing techniques in all aspects as several experimental factors such as number of samples, sampling method, fitness function, parameter setting, etc. were unknown for the existing techniques. To conclude, the overwhelming performance of HG- GA SVM with respect to various performance metrics was due to the exploitation of hyper clique

property for the generation of the initial population, which consists of sufficient number of 1s to maximize the fitness function of GA within few iterations.

5. Conclusion

This paper presents an efficient intrusion detection framework using Hypergraph based Genetic Algorithm (HG – GA) technique for parameter setting and feature selection in support vector machine. A weighted objective function was introduced to the design of an efficient IDS with high detection rate and low false positive rate. Hyper – clique property of hypergraph was exploited to bring down the complexity of genetic algorithm during its search for the optimal solution. NSL-KDD, a benchmark intrusion dataset was used for experimentation and validation purposes. Experimentations were carried out under two scenarios (i) Classifiers trained with all features and (ii) Classifiers trained with feature subset obtained from feature selection technique. The empirical results obtained from the experiments show that HG – GA SVM perform better than the existing techniques with respect to various performance metrics. HG – GA was found to be scalable, adaptive, robust and applicable over a wide range of data analytic problems in image processing, trust management, metadata quality analysis, stock market analysis and so on.

Acknowledgements

The first author thanks the Tata Consultancy Services for their financial support. The second and fourth author thanks the Department of Science and Technology, India for INSPIRE Fellowship (Grant No: DST/INSPIREFellowship/2013/963) and Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions (SR/FST/ETI-349/2013) for their financial support. The third author thanks the Department of Science and Technology—Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions Government of India (SR/FST/MSI-107/2015) for their financial support.

References

- [1] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, W.-Y. Lin, Intrusion detection by machine learning: A review, *Expert Syst. Appl.* 36 (2009) 11994–12000. doi:10.1016/j.eswa.2009.05.029.
- [2] Y.Y. Chung, N. Wahid, A hybrid network intrusion detection system using simplified swarm optimization (SSO), *Appl. Soft Comput.* 12 (2012) 3014–3022. doi:10.1016/j.asoc.2012.04.020.
- [3] Cyber Attacks on U.S. Companies in 2014, (n.d.). <http://www.heritage.org/research/reports/2014/10/cyber-attacks-on-us-companies-in-2014>.
- [4] Russia accused of unleashing cyberwar to disable Estonia | World news | The Guardian, (n.d.). <https://www.theguardian.com/world/2007/may/17/topstories3.russia>.
- [5] Georgia President's web site under DDoS attack from Russian hackers | ZDNet, (n.d.). <http://www.zdnet.com/article/georgia-presidents-web-site-under-ddos-attack-from-russian-hackers/>.
- [6] K. Scarfone, P. Mell, Guide to intrusion detection and prevention systems (idps), NIST Spec. Publ. (2007). http://ecinetworks.com/wp-content/uploads/bsk-files-manager/86_SP800-94.pdf (accessed December 5, 2016).
- [7] C. Koliass, G. Kambourakis, M. Maragoudakis, Swarm intelligence in intrusion detection: A survey, *Comput. Secur.* 30 (2011) 625–642. doi:10.1016/j.cose.2011.08.009.
- [8] F. Kuang, W. Xu, S. Zhang, A novel hybrid KPCA and SVM with GA model for intrusion detection, *Appl. Soft Comput.* 18 (2014) 178–184. doi:10.1016/j.asoc.2014.01.028.
- [9] S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, *Expert Syst. Appl.* 35 (2008) 1817–1824. doi:10.1016/j.eswa.2007.08.088.
- [10] L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li, B. Yang, D. Liu, Evolving support vector machines using fruit fly optimization for medical data classification, *Knowledge-Based Syst.* 96 (2016) 61–75. doi:10.1016/j.knosys.2016.01.002.

- [11] X. Gan, J. Duanmu, J. Wang, W. Cong, Anomaly intrusion detection based on PLS feature extraction and core vector machine, *Knowledge-Based Syst.* 40 (2013) 1–6. doi:10.1016/j.knosys.2012.09.004.
- [12] H. Li, S. Guo, H. Zhao, C. Su, B. Wang, Annual Electric Load Forecasting by a Least Squares Support Vector Machine with a Fruit Fly Optimization Algorithm, *Energies.* 5 (2012) 4430–4445. doi:10.3390/en5114430.
- [13] W. Wang, X. Liu, Melt index prediction by least squares support vector machines with an adaptive mutation fruit fly optimization algorithm, *Chemom. Intell. Lab. Syst.* 141 (2015) 79–87. doi:10.1016/j.chemolab.2014.12.007.
- [14] F. Friedrichs, C. Igel, Evolutionary tuning of multiple SVM parameters, *Neurocomputing.* 64 (2005) 107–117. doi:10.1016/j.neucom.2004.11.022.
- [15] C.-L. Huang, J.-F. Dun, A distributed PSO–SVM hybrid system with feature selection and parameter optimization, *Appl. Soft Comput.* 8 (2008) 1381–1391. doi:10.1016/j.asoc.2007.10.007.
- [16] H. ling Chen, B. Yang, S. jing Wang, G. Wang, D. you Liu, H. zhong Li, W. bin Liu, Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy, *Appl. Math. Comput.* 239 (2014) 180–197. doi:10.1016/j.amc.2014.04.039.
- [17] S.M. Hosseini Bamakan, H. Wang, T. Yingjie, Y. Shi, An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization, *Neurocomputing.* 199 (2016) 90–102. doi:10.1016/j.neucom.2016.03.031.
- [18] C.-L. Huang, C.-J. Wang, A GA-based feature selection and parameters optimization for support vector machines, *Expert Syst. Appl.* 31 (2006) 231–240. doi:10.1016/j.eswa.2005.09.024.
- [19] S. Sarafrazi, H. Nezamabadi-pour, Facing the classification of binary problems with a GSA-SVM hybrid system, *Math. Comput. Model.* 57 (2013) 270–278. doi:10.1016/j.mcm.2011.06.048.
- [20] K. Lin, Y. Huang, J. Hung, Y. Lin, Feature selection and parameter optimization of support vector machines based on modified cat swarm optimization, *Int. J. Distrib.* (2015).
- [21] M. Zhao, C. Fu, L. Ji, K. Tang, M. Zhou, Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes, *Expert Syst. Appl.* 38 (2011) 5197–5204. doi:10.1016/j.eswa.2010.10.041.
- [22] Z. Chen, T. Lin, N. Tang, X. Xia, Z. Chen, T. Lin, N. Tang, X. Xia, A Parallel Genetic Algorithm Based Feature Selection and Parameter Optimization for Support Vector Machine, *Sci. Program.* 2016 (2016) 1–10. doi:10.1155/2016/2739621.
- [23] J. Kamruzzaman, R. Begg, Support vector machines and other pattern recognition approaches to the diagnosis of cerebral palsy gait, *IEEE Trans. Biomed.* (2006).
- [24] Z. Qi, B. Wang, Y. Tian, P. Zhang, When Ensemble Learning Meets Deep Learning: a New Deep Support Vector Machine for Classification, *Knowledge-Based Syst.* (2016).
- [25] M. Pontil, A. Verri, Support vector machines for 3D object recognition, *IEEE Trans. Pattern Anal.* (1998).
- [26] F. Melgani, L. Bruzzone, Classification of hyperspectral remote sensing images with support vector machines, *IEEE Trans. Geosci.* (2004).
- [27] Christiannini, N., and J. Shawe-Taylor. "Support vector machines and other kernel-based learning methods." (2000)., (n.d.).
- [28] Davis, L. (1991). *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold, (n.d.).
- [29] K. Kannan, B.R. Kanna, C. Aravindan, Root Mean Square filter for noisy images based on hypergraph model, *Image Vis. Comput.* 28 (2010) 1329–1338. doi:10.1016/j.imavis.2010.01.013.
- [30] M. Raman, K. Kannan, S. Pal, Rough Set-hypergraph-based Feature Selection Approach for Intrusion

Detection Systems, Def. Sci. (2016). <http://publications.drdo.gov.in/ojs/index.php/dsj/article/view/10802> (accessed December 5, 2016).

- [31] N. Somu, K. Kirthivasan, S.S. V.S., A computational model for ranking cloud service providers using hypergraph-based techniques, *Futur. Gener. Comput. Syst.* 68 (2017) 14–30. doi:10.1016/j.future.2016.08.014.
- [32] N. Somu, M.R.G. Raman, K. Kirthivasan, V.S.S. Sriram, Hypergraph Based Feature Selection Technique for Medical Diagnosis, *J. Med. Syst.* 40 (2016) 239. doi:10.1007/s10916-016-0600-8.
- [33] C. Berge, E. Minieka, *Graphs and hypergraphs*, 1973. <http://tocs.ulb.tu-darmstadt.de/10727930.pdf>.
- [34] NSL-KDD Data Set, [Online]. Available: <http://nsl.cs.unb.ca/NSL-KDD/>, (n.d.).
- [35] M. Tavallaei, E. Bagheri, W. Lu, A detailed analysis of the KDD CUP 99 data set, *Proc.* (2009). <http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=15084639> (accessed December 5, 2016).
- [36] C.-C. Chang, C.-J. Lin, LIBSVM, *ACM Trans. Intell. Syst. Technol.* 2 (2011) 1–27. doi:10.1145/1961189.1961199.
- [37] Witten, Frank, Eibe, *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition, (n.d.).
- [38] R. Singh, H. Kumar, R.K. Singla, An intrusion detection system using network traffic profiling and online sequential extreme learning machine, *Expert Syst. Appl.* 42 (2015) 8609–8624. doi:10.1016/j.eswa.2015.07.015.
- [39] E. de la Hoz, A. Ortiz, J. Ortega, E. de la Hoz, Network Anomaly Classification by Support Vector Classifiers Ensemble and Non-linear Projection Techniques, in: *Springer Berlin Heidelberg*, 2013: pp. 103–111. doi:10.1007/978-3-642-40846-5_11.
- [40] C.-H. Tsang, S. Kwong, H. Wang, Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection, *Pattern Recognit.* 40 (2007) 2373–2391. doi:10.1016/j.patcog.2006.12.009.
- [41] H. Gunes Kayacik, A. Nur Zincir-Heywood, M.I. Heywood, A hierarchical SOM-based intrusion detection system, *Eng. Appl. Artif. Intell.* 20 (2007) 439–451. doi:10.1016/j.engappai.2006.09.005.