1. **Introduction**

“Digital World” – an epitome of digitization and its remarkable impact on the way in which communication across the globe has transformed which make way for new technological advancements and challenges. The plethora of security breaches in the digital ecosystem makes it more of a concern. Several researchers’ focuses in the development of techniques like firewall, anti-virus, access control, encryption standards etc. to safeguard their critical network infrastructure and information systems from the hands of the attackers. However, these techniques seems to be immature as the nature of attackers are changing dynamically and malware activities become more sophisticated in nature. This problem is evident due to the various security breach incidences happening globally. WannaCry, the ongoing ransomware program and a worm, has hit around 230,000 computers in 150 countries exploiting a vulnerability in the Windows Operating System. In September 2016, the email service provider Yahoo, witnessed the biggest data breach in history compromising sensitive information including hashed passwords of around 500 million customers. Moreover, a report from FBI in June 2016 confirms a 300% increase in the number of ransomware attacks over 2015.

According to NIST “*Intrusion is an attempt to compromise Confidentiality, Integrity and Availability (CIA) or to bypass the security mechanisms of a computer or a network and Intrusion detection is the process of monitoring the events occurring in a computer system or network and analysing them for signs of intrusions, defined as attempts to compromise the CIA or to bypass the security mechanisms of a computer or network. Intrusion Detection Systems (IDSs) are software or hardware products that automate this monitoring and analysis process*”. Generally on basis of attack detection method, IDS can be categorized into two namely (i) Signature based IDS (ii) Anomaly based IDS. The former type is trained with the patterns or knowledge obtained from the known attacks whereas the latter is trained with the behaviour of the normal users. Though signature based IDS consists of higher detection rate, but it cannot contribute towards the zero day attack since it requires the frequent updation of signature database. Hence the majority of researchers focus mainly on anomaly based IDS since it possesses the ability of identifying the novel attacks. However the major drawback of anomaly based IDS is the higher false alarm rate due to the existence of massive, high dimensional and ill-conditioned network traffic dataset. Thus the development of a robust and efficient Intrusion detection system still remains an open research problem.

In recent years, intrusion detection is considered to be a classification problem (distinguishing the normal and malicious patterns) for which various machine learning and statistical techniques like Support vector machine (SVM), Artificial Neural Networks (ANN), Decision tree, K- Nearest Neighbour (KNN), Rough Set theory (RST), Bayesian theory, Principle Component Analysis (PCA) etc. were utilized extensively. Among these aforementioned tools, support vector machine and its variants was found to be a predominant choice of the researchers for developing a robust, efficient, adaptive IDS due to its characteristics like higher generalization ability, structural risk minimization, resist to over fitting issues etc. However according to Shih-Wei Lin et al & Cheng-Lung Huang et al there exists two significant drawbacks in traditional SVM model namely (i) Proper model parameter setting (ii) Curse of Dimensionality since they both have much contribution towards the model’s complexity and classification accuracy. In order to address these problems several research works were carried out which predominantly focus on meta-heuristic techniques like Genetic algorithm (GA), Particle swarm optimization (PSO), Fruit fly optimization (FOA) etc. as reported in table 1.

**Table 1** – Related Works

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Authors | Parameter  Optimization | Feature  Selection | Technique | Evaluation Criteria | Application |
| Yi Lui et al |  |  | Filtered and supported sequential forward search | Classification accuracy | Generic applications |
| Susana M. Vieira et al |  |  | Modified Binary Particle Swarm Optimization | Classification accuracy | Mortality prediction of septic patients |
| Liming Shen et al. |  |  | Fruit fly optimization technique | Classification accuracy | Medical applications |
| Seyyid Ahmed et al |  |  | Binary Dragon Fly Algorithm | Classification accuracy rate, number of selected genes, computational time | Cancer diagnosis |
| XiaoLi Zhang et al |  |  | Ant Colony Algorithm | Diagnosis accuracy | Intelligent Fault Diagnosis of Rotating Machinery |
| Mingwei Wang et al |  |  | Modified Binary Ant Colony Optimization  Algorithm | Classification accuracy (Fitness value of objective function) | Remote sensing image classification |
| Shish-Wei Lin et al |  |  | Particle Swarm Optimization | Classification accuracy | Generic application |
| Cheng-Lung Huang et al |  |  | Continuous valued PSO- Parameter Optimization  Discrete-valued PSO- Feature selection | Classification accuracy | Load balancing in distributed systems |
| Hui Ling Chen et al |  |  | Parallel time variant Particle Swarm Optimization | Classification accuracy | Generic application |
| Cheng-Lung Huang et al |  |  | Genetic Algorithm | Area under the ROC curve | Generic application |
| Fangjun Kuang et al |  |  | Genetic Algorithm – Parameter Optimization  Kernel principal component analysis – Feature selection | Classification accuracy | Intrusion detection system |
| Soroor Sarafrazi et al |  |  | Continuous - valued Gravitational Search Algorithm – Parameter Optimization  Discrete - valued Gravitational Search Algorithm (GSA) – Identification of the optimal feature subset | Classification accuracy | Generic application |
| Sebastian Maldonado et al |  |  | Sequential backward selection | Mean test error | Generic application |
| [Alaa Tharwat](http://www.sciencedirect.com/science/article/pii/S0167865516302720) et al |  |  | Bat Algorithm | Test error rate | Generic application |
| [Yunqiang Zhang](http://www.sciencedirect.com/science/article/pii/S0167865514003614) et al |  |  | Social Emotional Optimization Algorithm (SEOA) | Classification Accuracy | Generic application |
| [Xiaoyuan Zhang](http://www.sciencedirect.com/science/article/pii/S0925231214010248) et al |  |  | Bare bones differential evolution (BBDE) algorithm | Diagnosis accuracy | Fault diagnosis |

\*- Feature Extraction

# - Parameter Optimization and Feature Extraction

$ - Parameter Optimization

However, these heuristic functions do not guarantee the optimal solution with better convergence rate. Hence this research work attempts to improve the performance of the SVM model with a novel Hyper clique based improved Binary Gravitational search algorithm for the development of efficient IDS. Binary Gravitational search algorithm is a novel optimization algorithm proposed by E Rashedi et al inspired from the Newton’s law of gravity. It has proven its efficiency in terms of better convergence, guaranteeing the global optima, computationally attractive etc. in wide variety of applications such as. The major contributions of this paper are

1. Hyper clique property of hypergraph is exploited in the generation of initial population of BGSA which enhances its performance in terms of minimal time complexity and prevents the pre-mature convergence.
2. Newton - Raphson inspired statistical function is introduced in BGSA to minimize the trade-off between the exploration and exploitation of agents.
3. The proposed HC-IBGSA is adopted to improve the performance of SVM classification model through identifying the optimal model parameters and informative feature subset.
4. The two standardized benchmark dataset namely KDD cup 1999 intrusion detection dataset, NSL-KDD cup dataset along with the synthetic intrusion detection dataset obtained from SASTRA University is used for the experiments and validation purpose. The performance of the proposed HC-IBGSA SVM IDS model is evaluated through the various performance metrics like classification accuracy, detection rate, false alarm rate and runtime analysis.

**2. Materials and Methods:**

**2.1 Support Vector Machine (SVM):**

SVM is statistical theory based machine learning technique which addresses various classification and regression problems across a wide variety of applications like Image processing, Medical diagnosis, Remote sensing etc. It relies on Structural risk minimization principle which makes its performance better in comparison with other machine learning techniques like artificial neural networks, Bayesian theory 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”. The main idea behind the SVM is that, all data samples are viewed as points in a n-dimensional space and they are separated (classified) through the construction of hyperplanes. A better performance can be experienced from the SVM for linearly separable samples, however for the case of non-linear separable dataset kernel functions are used to map them in a higher order dimensions, from which the dataset are linearly separable. The main challenge in the SVM is choosing a suitable kernel function and the identification of optimal kernel parameters. In this section, we discuss few basic definitions and concepts behind the SVM for the classification of both linear and non-linear separable datasets.

**Class +1**

**Class -1**

Optimal Hyperplane

Margin

Support Vectors

**Figure 1**: Linear SVM

**2.1 Linearly Separable data:**

Consider a binary classification problem with training datasets and S samples, represented by n attributes. Each training samples belongs to either one of the two classes. The classification function of SVM or the decision boundary which separates the data points of both positive and negative classes are

On combining (1) & (2) we get

where is the weight vector normal to hyperplane, b is the bias value and x is the input sample. From the Figure 1 it is clear that the distance between the two planes (margin) is and SVM attempts to maximize it wrt to the condition

under the constrain. This can be viewed as quadratic optimization problem which can be solved by Lagrange function shown in eqn (3)

Where is the Largrange multiplier. Thus in order to obtain the optimal weight and bias values, the eqn has to differentiated wrt to **W** and b and equated to

On substituting the value of () & () in () we get a dual Largrangian

Under the constrain,

The above equation() has to be maximized with respect to non-negative to determine the values of In SVM, geometrically the training sample having non zero values are called as support vectors and they are the points that lie closer to the hyperplane (Fig). Thus if a unknown sample has to be classified, it is enough to obtain the sign of equation(). If the y value is positive, then the unknown sample belongs to +1 class or else vice versa.

**2.1.2 Non-Linear SVM Classifier:**

For the case of the non-linear separable data samples, the same concept is extended by introducing a stack variable in eqn () ().

On combining () & () we get

The objective function of the SVM can be modified as

Subjected to

where C is the non-negative, user defined penalty factor which minimize the trade-off between the margin width and slack variable penalty. In case of non-linear SVM, the data samples are mapped to a higher dimensional plane through a non-linear function called kernel function. Thus the equation () can be altered as

Under the constrain,

where is the kernel function and most commonly used kernel functions in SVM are listed in table 2 . In this study we have applied radial basis kernel function in which C and values are optimized to obtain a better performance in SVM.

**Table 2**: Kernel Functions

|  |  |
| --- | --- |
| Kernel | Function |
| Linear |  |
| Polynomial Function |  |
| Radial Basis Function |  |
| Sigmoidal Function |  |
| Fourier Kernel Function |  |

**2.2 Binary Gravitational Search Algorithm (BGSA)**

The Gravitational search algorithm is a novel stochastic search technique proposed by Rashedi et al inspired from the Newton law of gravitation and motion. In GSA each agents are considered as objects and they are evaluated through their masses. All objects are attracted towards each other through a gravitational force causing a global movement of objects towards the heavier object (Best solution).

The GSA algorithm can be summarized as follows

Step 1: *(Initialization)* consider K objects and position of those objects are initialized randomly. Let the position of object be in the n dimensional plane can be defined as

Step 2: *(Mass of each objects)* The mass of each objects are evaluated by the fitness function. The mass is computed through the eqn ()

For minimization problem, ; =

Step 3: *(Force)* The force exerted by object i on object j is computed through the eqn()

Where is the gravitational constant, is the mass of the object i and object j respectively, is the Euclidean distance between object i & object j, is small valued constant. Thus the total force exerted by each objects on others are

Where is uniform random variable in the interval [0 1]

Step 4: *(Acceleration)* The acceleration of the object i at the time t can be computed as

Step 5: *(Velocity and New position)* The updated velocity and position of each objects are calculated using eqn()

Where is the random variable with the interval [0 1].

As most of the optimization technique can also operates over the binary search space, the binary version of GSA is proposed by Rashedi et al, where each dimension in the search space can take either 1 or 0. The displacement of the objects ie change of position is also restricted to 0 or 1. The steps involved in BGSA is similar to that of GSA except few minor changes in step 5. The eqn () can be modified as

In order to minimize the randomness in while updating the position of the objects in the next iteration ie to avoid the premature convergence, we have proposed a novel statistical method. According to that, the eqn () can be altered as

**3.1 Hypergraph:**

Hypergraph is a mathematical framework and generalization of a traditional graph theory through which the n-ary relationships among the variables (Objects) can be represented more significantly. Hypergraph based computational models has proven its dominance in various applications like Image processing, Cyber intelligence, Cloud computing, Bio-informatics etc in terms of minimal time complexity. In the section we discuss few basic definitions of hypergraph and clique property which can be hybridized with Gravaitiaonal search algorithm for its improved performance.

**Definition 1**: *(Hypergraph)* Consider be finite set of vertices, then the hypergraph is the family where E is the hyperedges such that .

**Definition 2**: *(Representative Graph)* Consider a hypergraph, then the representative graph of H is a graph such that

1. , when H has no repeated hyperedges
2. , if and only if

**Figure** : Hypergraph

**Definition 3**: *(Closed Neighborhood)* Consider a graph and let , then the closed neighborhood of m in is defined as follows

**Definition 4**: (Complete graph): Consider a undirected graph where represent the set of vertices and represents the set of edges. The Graph G is said to be compete if there exists an edge between the vertices

**Definition 5:** (K-Clique): For a given graph, K-Clique of G is a subset , such that sub graph C satisfy the definition 4. (a)3-Clique

(b) 4-Clique (c) 5- Clique

**Figure**

**Proposition 5.1:** Considera graph and clique , then C is the clique of G if and only if

Proof: Let C be the clique of graph G. Let, then and . Conversely, consider. Let a, b be a distinct elements of C, then by hypothesis, a and b are adjacent. Hence C is the clique of   
G.

**3.HG GA – SVM: The Proposed Methodology**

In this section, we discuss about the design of effective intrusion detection system through the optimal SVM which utilize the HG-IBGSA technique for parameter setting and feature subset selection. The overall objective of the proposed approach is to have maximized detection rate and minimized false alarm rate along with the optimal number of features. The main procedure in HC-IBGSA SVM as follows,

***(i)Generation of initial population:***

In the traditional BGSA, the position of each objects are generated randomly and they are evaluated through the fitness function from which the mass, force, acceleration, updated velocity & position are computed as discussed in section. Similarly in the proposed HG-IBGSA the generation of initial population has the following phases (i) Random generation of two dimensional matrix of order M X K where M & K is the number and size of each objects respectively (ii) Representation of each bit in the string (objects) as vertices (iii) Building the hyperedges to construct the hypergraph by inducing the hyperedge relation between the each objects and strings (iii) Invoking the hyper clique property among the data for inducing the optimal number of ‘1’ in each objects and minimizing the hyper relations among the objects. (Fig)

In HG-IBGSA, each objects (binary representation) comprises of three parts representing kernel parameters & feature subset respectively (Fig). The code length corresponds to the number of bits used to represent values and similarly corresponds to the total number of features in the dataset.

**IDS Dataset**

Data Mapping

Data Normalization

Data Pre-processing

Training Dataset

Testing Dataset

Feature selection

Training SVM

Training SVM

Termination Condition?

*Optimized Kernel Parameters and Feature Subset*

Random generation of population

Hypergraph Construction

Application of Hyperclique Property

Kernel Parameter

Feature Selection

Compute Mass (Eqn()), Force (Eqn()), Acceleration (Eqn()), Velocity (Eqn()), Posistion (Eqn())

Conflict?

Update the Position (Eqn())

**Yes**

**No**

**Yes**

**No**

**Figure** : Hypergraph and Binary Gravitational Search Algorithm (HG – BGSA) based Technique for Parameter Setting and Feature Selection in Support Vector Machine

***(ii)Definition of fitness function:***

In this work, the objective function is designed by considering the predominant performance metric like detection rate and false alarm rate which decides the effectiveness of any IDS. In addition to these metrics, number of features are also considered since it has a greater impact towards the complexity of the learning model. Thus it becomes a multi criteria decision making (MCDM) approach which can be solved by the weighted objective function as shown in eqn().

Where ,, are the predefined weights of detection rate, false alarm rate, number of features respectively. Among all these, is set to a maximum value since the detection rate has to be increases and is set to a least value has the false alarm rate should be minimized. The calculating procedure of detection rate and false alarm rate were reported in table

|  |  |
| --- | --- |
| **Performance metric** | **Equation** |
|  |  |
|  |  |

**HC-IBGSA SVM:**

The working of HC-IBGSA is made up of two phases namely (i) Parameter setting for the RBF kernel function (ii) Optimal feature subset selection. The workflow of the proposed approach as follows:

1. ***(Generation of training & testing dataset)*** From the given input dataset , generate the training and testing dataset in the ratio 80:20 such that
2. ***(Population Generation)*** Generate the initial population as discussed in section. The binary vector which represents the value are converted to a floating point value using the eqn ()

Where is the lower and upper bound of the kernel parameter, CL is the code length and d is the decimal value of the binary string. Similarly for the feature selection, the presence of ‘1’s or ‘0’s in the binary vector representing the feature subset corresponds to the presence or absence of the feature respectively.

Step 3: ***(Training SVM)*** During the training phase of the SVM, the kernel parameters and training dataset consisting only of identified features by each objects(agents) are involved in the training process.

Step 4: ***(Testing SVM)*** In the testing phase, the performance of each individually trained SVM is evaluated in terms of detection rate, false alarm rate and number of features using the fitness function discussed in section

Step 5: ***(Termination condition)*** Once the performance of each objects (Agents) is evaluated during the testing phase, the HC-IGSA SVM will returns the optimal kernel parameter values and informative feature subset if the termination condition(maximum fitness value or generation) is satisfied. If not go to the step 6.

Step 6: ***(Generation of new positions)***If the termination criteria is not meet, then generation of new positions of each objects is carried by the computing its mass eqn(), force eqn(), acceleration eqn(), velocity eqn() and position eqn(). However due to rapid reduction of diversity, there may be chance of algorithm to trap in local optima (Conflict). In order to avoid the conflict, the gobal best solution for each iteration is monitored and if it remains the same, the eqn () is included while updating the position of each objects[].

Where is the total number of population, is the random number [-1,1]. The above equation computes the mean position of top k best objects and update the position of each objects thereby better position adjustment is carried out.