DOI: http://dx.doi.org/10.26483/ijarcs.v8i9.4857

ISSN No. 0976-5697

Volume 8, No. 9, November-December 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A FEATURE SELECTION TECHNIQUE FOR INTRUSION DETECTION SYSTEM BASED ON IWD AND ACO

Farha Haneef
Research Scholar
NITTTR, Bhopal, India

Shailendra Singh,
SeniorMember, IEEE
NITTTR, Bhopal, India

Abstract: Rapid advancements in the internet technology and its vulnerabilities have led researchers to devise intelligent systems that can provide network security. Intrusion detection systems (IDS) scrutinize all the features to detect intrusive data. Some of the features may be redundant or irrelevant to the detection process, which results in computational complexities and increased training time. To mitigate this problem, a process known as feature selection is used to remove the redundant and irrelevant attributes of the dataset. This paper proposes an intelligent hybrid technique for the purpose of feature selection of KDD CUP'99 dataset using the concepts of metaheuristic optimization. This hybrid approach combines the concepts of Intelligent Water Drops(IWD) and Ant Colony Optimization(ACO)to select features from data. The objective of this work is to optimize the process of feature selection in order to achieve optimal feature subset and reduce the training time.

Keywords: Intrusion Detection System,; Feature Selection; Ant Colony Optimization; Intelligent Water Drops; Metaheuristic Optimization.

1. INTRODUCTION

Intrusion Detection System is an ideal tool used widely for network security at the organizational, institutional and private workspaces level. It can work as a software or a hardware component. IDS is normally placed before a network of networks or a simple host depending on the requirement. The basic working of IDS is to scan the incoming network traffic and to identify any deviant behaviour of a network data packet from what is defined as normal for the safety of the system under consideration [1]. This process is known as outlier detection and the performance of an IDS is majorly judged by its efficiency in outlier detection. Outlier detection is the simple act of identification of normal data from intrusive data.

The building of an IDS model usually consists of two processes- feature selection process and classification process. IDS can be mainly assumed as a classification system, since its work is to simply classify normal from intrusive data packets, but classification alone is not sufficient for an effective IDS model[2]. Many researchers in the past have established the fact that feature selection process is a must before classification process in IDS to improve classification accuracy[15].

Feature Selection becomes an important part of IDS construction because traffic analysis becomes difficult due to large number of features in the audit data[3]. The large number of feature space normally contains redundant and irrelevant data samples. Presence of noisy features often results in high false positive rate which means highly erroneous results which cannot be accepted when security of a system is under consideration. The large feature space containing duplicate attribute values also increases the computational cost of the system in terms of high training and testing time, large chunks of memory space utilization and delay in effective decision making[4]. Hence feature selection process becomes an extremely important step for designing IDS model.

This paper discusses the process of feature selection in Intrusion Detection Model. The need for feature selection in IDS model is to find the optimal feature subset from feature space to reduce the training time. The recent advancements in the field of metaheuristic optimization show that algorithms of this field generate most optimal solutions, because these methods use the knowledge of past experiences as well as converge to the solution in a computationally efficient manner.

In the proposed work, a unique hybrid algorithm for the feature selection process for IDS model is being suggested. This algorithm combines the concepts of Intelligent Water Drops(IWD) and Ant Colony Optimization algorithm. The IWD has been used to generate a collection of partial solutions which are then passed to the ACO algorithm. The ACO optimizes the solutions and finds the best solution among them.

The rest of the paper is designed as follows. Section II contains some related work on the feature selection methods of IDS. Section III contains the Proposed Methodology for the research. Section IV contains the Experimental Results and finally the work is concluded.

2. RELATED WORK

Feature Selection Process in the field of Intrusion Detection System is the most sought after field of research in recent times. The researchers have worked in all the methods ranging from filter methods to wrapper methods involving techniques from data mining, machine learning various intelligent optimization techniques. The newer inventions are being done in the field of metaheuristics which are being preferred due to their capability of producing most optimal solutions in least computational time complexity.

A two grains levels network intrusion detection system was proposed by Safaa O. Al-mamory et.al in [5] which used a very fast decision tree algorithm. The method reduced 41 features of KDD CUP'99 dataset to 20. Information gain was used to reduce the number of features by ranking method.

Those features which had information gain value more than the average of the dataset (IG average=0.22) were selected into the subset. Training the model using reduced 20 feature subset lowered the processing time to approximate 3x10-6 sec per example.

Sung and Mukkamala[6] proposed a method for feature selection using Support Vector Machine (SVM). SVM is one of the methods based on Statistical Learning Method mainly used for classification purposes. The feature selection starts with taking the whole feature set as input and then analyzing its performance by successively removing one feature at a time. If removal of a feature leads to a better performance by the SVM classifier, then it is not included into the subset. The number of features selected by this process is 13 out of 41.

An innovative algorithm for feature selection by using tuple selection and attribute selection was proposed by Sannasi Ganapathy et.al in[7]. Information gain ratio and rules were used for attribute selection. This agent based attribute selection model resulted in a reduction of 41 features of KDD CUP'99 dataset to 19 features.

Another work where Rough Set Theory(RST) has been used for feature selection is proposed by Rung-Ching Chen et.al in [8] for network intrusion detection. This method involves creating a decision or an information table which holds the description of features of processes. Grouping of similar types of attributes together is done depending upon the indiscernible relation existing for each feature. This phenomenon is utilized to create a minimal subset of attributes. 29 features out of 41 features of DARPA data was selected from this method. Principal Component Analysis is one of the most popular methods for feature selection. It is a statistical method very popular for projecting lower dimensional feature value.

Heba, F.Eid in [9] has proposed a method for feature selection for anomaly intrusion detection system using PCA method. The principal components are selected on the basis of eigenvalues. The features having values more than a certain threshold where D represents the dimensionality of the original input dataset are selected. NSL-KDD dataset having 41 data features are used for the experimentation. The proposed PCA method was able to select 23 features out of 41 total features. The dimensionality was considerably reduced by 56%. The training time and testing times also got reduced due to dimensionality reduction. The training time and testing time for the datasets got sufficiently reduced to 1.7 ms and 1.3 ms respectively with the PCA method.

B.M. Aslahi Shahri in [10] have proposed a wrapper method for feature selection for IDS model. This wrapper model combines the concept of Genetic Algorithm(GA) with Support Vector Machine(SVM). GA is used to generate partial feature subsets which are evaluated for their goodness by the performance of SVM classification on it. The kernel function of the SVM classifier is used as the fitness function for the partial solutions, where its detection rates are used as the fitness values. This wrapper method was able to achieve 10 features out of total 45 features.

Ant Colony optimization is a meta-heuristic algorithm that generates optimal solutions in an evolutionary manner. Ant Colony Optimization(ACO) and K-NN classifier were combined in a wrapper method for feature selection by Mehdi Hosseinzadeh, Aghdam et.al in [3]. In this method, ACO has been used for candidate feature subset selection and the evaluation of the goodness of this candidate subset is done by the K-NN classifiers. Here the problem of selecting the most feasible feature into subset is modeled as a problem of best path selected by an ant to traverse from its colony to its food source. The feature subset selected was a maximum of 8 for Probe and a minimum of 3 for R2L class.

Due to this reduction, the detection error for the IDS model was reduced by 24%. It can be very easily understood from the trends of feature selection techniques used by different researchers that there has been a considerable shift from using exhaustive search techniques to using intelligent optimization or evolutionary methods for this purpose. And this observation has only inspired this research work which aims to use metaheuristic optimization for feature selection in a unique way by combining two algorithms namely Intelligent Water Drops(IWD) and Ant Colony Optimization(ACO) as discussed in next section.

3. PROPOSED METHODOLOGY

The purpose of this research is to develop a unique optimized method for feature selection for an IDS model. The earlier researches done in the field of intrusion detection have clearly stated the importance of the feature selection process in IDS. The latest trends in these works show an interest in the field of metaheuristic optimization algorithms due to their fast convergence properties which reduces the overall processing costs of the design. Taking inspiration from these approaches, we here in this work propose a hybrid technique of feature selection combining the concepts of two optimization techniques. The usage of two optimization techniques have been done to reduce the overall computation cost as against the existing models. Hence, this proposed work, presents a hybrid algorithm for feature selection for an intrusion detection system. This model combines IWD with ACO. The flow of the proposed work is given by the Figure 1 given below.

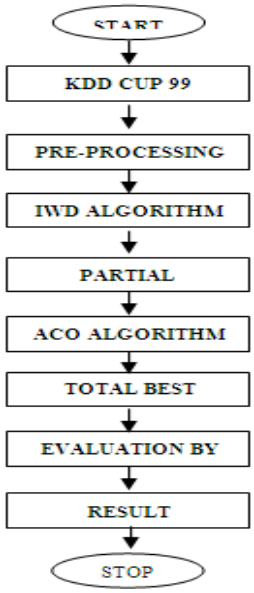

Figure 1. Proposed Model for IDS

The proposed model brings together two concepts of metaheuristic optimization where one algorithm tries to find the best solution from a partial solution developed by another

algorithm. Here Intelligent Water Drops algorithm has been used to generate the partial solution. This partial solution is optimized by the other algorithm Ant Colony Optimization to create the final best solution in terms of a Feature subset of minimal length which reduces the training time of the model. The description of the process is as follows-

A. Intelligent Water Drops Algorithm

Intelligent Water Drops(IWD) is an intelligent metaheuristic optimization algorithm developed by researcher Hamed Shah Hosseini[11]. It is a nature inspired algorithm which derives its concept of evolution from the mannerisms in which water drops of a river flow through their path. The naturally occurring behavior of water drops to cast a way for them has been beautifully conceptualized into an algorithm.

The path through which the river travels always has some soil in it. When a water drop travels through an area, it has a velocity and carries some content of soil swiped from that area. Hence, that area becomes deprived of that much soil as carried by the waterdrop and the chance of other water drops to take that path increases. This leads to more lessening of soil from that area. Hence, it becomes a natural concept, that water drops tend to prefer that path which has lesser soil in it, the velocity of the drop increases during this path, and while they cross, the drops carry away more soil from it. The optimal solution for this algorithm is developed in terms of best path travelled by various water drops[12].

The problem of optimization is viewed as a problem of graph, where E numbers of edges connect N number of nodes. The solution is formulated in terms of various paths travelled by different IWDs. One iteration is assumed to be completed when all IWDs create their own distinct paths and the best solution from these paths is chosen to be the one with minimum length. The iterations are repeated until a maximum number of iterations are achieved by it. The algorithmic description of the whole process is as given below-

a) Pre-Requisites-

The feature selection problem is represented as a graph where the attributes of KDD CUP'99 dataset is taken as input. The dataset contains 41 features in all. Each feature makes a node of the graph and each node is connected to all the other nodes, i.e to say that all the features have a path connecting to each of the other feature.

Phase 1: Initialization In step initialization of the values of static and dynamic parameters is done. This step also produces the graph to be worked on:

(i) Static parameters : NIWD denotes the number of waterdrops. Variables (av, bv, cv) update the velocity of water drops. Variables (as, bs, cs).update the soil of the local path. MaxIter and initSoil are the maximum number of iterations and the initial value of the local soil, respectively.

(ii) Dynamic parameters :Dynamic parameters are initialized at the beginning of the iteration and updated during the search. These comprise f a list of features visited by each water drop k as VcIWDk. Velocity and soil of water dropk at the start of the search as initVelIWDK and SoilIWDK, respectively.

(iii) The complete problem is understood as a graph G(N,E).Where N is the number of nodes and E is the edges. The nodes denote the features for IDS problem. The graph is a complete graph and all its nodes are connected with each other. Water drops are randomly distributed on the nodes. Every drop k represents a distinct feature which is the source for the drop and is added to its visited list of features.

Phase 2: Solution building :This phase calculates the solution for every water drop for a single iteration. It comprises following steps:

(i) Selection of edge: A water drop k placed on a feature i selects the next feature j ,that has not yet been added to its visited list using the probability function as follows:

1. The selection of feature j, when a water drop k is residing at a feature i is done according to the following probability-

$$P_i^{IWD_k}(j) = \frac{f(soil(i,j))}{\sum_{1 \neq V_c} IWD_k f(soil(i,j))} \quad (1)$$

Where soil (i,j) represents the value of soil between the path connecting features i and j and the function f(soil(i,j)) returns inverse value of soil(i,j)

2. The updating of velocity of water drop k as it moves from feature i to j is done as-

$$\mathrm{Vel}^{\mathrm{IWDk}}(\mathrm{t}+1) = \mathrm{Vel}^{\mathrm{IWDk}}(\mathrm{t}) + \frac{a_v}{b_v + c_v\, soil(i,j)} \quad (2)$$

3. The soil carried away by the IWD during its flow is calculated as

$$\Delta soil(i,j) = \frac{a_s}{(bs + cs.time(i,j;\mathrm{Vel}^{\mathrm{IWDk}}(t+1)))} \quad (3)$$

where time (i, j) ; $\mathrm{Vel}^{\mathrm{IWDk}}$ (t + 1) is the time needed by the drop k to make a movement from node i to node j. It is represented as

$\mathrm{time}(i,; IWDk(t+1) = HUD(j)/\, velIWDk\ (t+1)$

4. The soil updating on the path where IWD travels is done as follows-

$$soil(i,j) = (1-\rho n).soil(i,j) - \rho n.\Delta soil(i,j) \quad (4)$$

$$soil_{IWD} = soil_{IWD} + \Delta soil\ (i,j) \quad (5)$$

Phase 3: Restructuring From all the solutions found out by every IWD, the iteration's best solution *T* IB is calculated as follows:

1. Call the ACO algorithm, and from all the solutions of IWD generated, select the best solution as-

$$T^{\mathrm{IB}} = S(t) \quad (6)$$

2. For the path, where the iterations best solution is found is updated as-

$$\mathrm{soil(i,j)} = \left(1+\rho_{\mathrm{IWD}}\right).\mathrm{soil(i,j)} - \rho_{\mathrm{IWD}}\frac{1}{q(T^{IWD})} \quad (7)$$

Phase 4: Termination condition Phase 2 and 3 are repeated

until the maximum number of iterations is reached. For every iteration, T^{TB} gets replaced by T^{IB}, if the latter is better than the former as shown.

$$T^{TB} = \begin{cases} T^{TB} \text{ if } q(T^{TB}) \geq q(T^{IB}) \\ T^{IB}, \text{otherwise} \end{cases} \quad (8)$$

If iterations best solution is found to be better than the total best solution then it replaces the latter.

B. Ant Colony Optimization Algorithm(ACO)

Ant Colony Optimization is another evolutionary metaheuristic optimization algorithm founded by Marco Dorigo in 1992[13]. ACO is an algorithm that derives inspiration from the behavioural patterns of ants who seek a path between their colony and the food source. The ants in their natural environment have a tendency to wander randomly until they find a food source. When they find the food, they return to their colonies leaving pheromone trails on the way back. If the other ants find the same trail , they very likely follow the same path and reinforce the trail, if they find food. This pattern makes the bunch of ant to select the closest path between their colonies and food source[14]. The behavioural pattern of the ant is presented algorithmically as follows-

The ACO algorithm performs the process of feature selection by initializing an empty set and depending upon some evaluation, adding the feasible features into the optimal set. For the proposed work IWD-ACO, the algorithm starts, when IWD has found its partial solution. The ACO algorithm evaluates all the solutions, and returns the best among them.

a) The working starts with ants being distributed randomly over the partial solutions developed by the IWD.

b) For each ant belonging to k=1 to m, where m is the cardinality of the partial solution generated by an IWD, an empty list to store the traversal path is initialized as Sk(t).

c) The ant k then traverses to add next possible feature i at a time step t as follows using a probability function which is a trade off between the pheromone intensity and heuristic information for feature i.

d) The next step involves the Pheromone updating process. After every ant has generated a completed solution, evaporation of pheromone on all nodes is done. Then each ant deposits a quantity of pheromone on every node that it visited.

(iii) Heuristic information

A heuristic is technique developed for finding a solution of a problem, where classical methods are either too slow or are not capable of doing so. A heuristic function is a function which evaluates the goodness of a solution on the basis of past experience. In exhaustive search algorithms it guides in order to get optimal solution. The proposed work also makes use of heuristic evaluation during the feature selection phase. This evaluation is used to evaluate the goodness of the next feature to be selected in a subset.

(a) In IWD, the heuristic to determine, which path to follow next or which feature to select next during search phase are two metrics— one is the content of soil on the path and the other is the HUD(heuristic undesirability).

1. Soil(i, j) denotes the content of soil on path connecting node i and node j. The lesser the soil, the greater will be the probability of selecting the path.

2. HUD(i, j) refers to the heuristic undesirability of selecting node j after selecting node i into the solution according to the problem concerned. For the problem of feature selection, we determine whether to select feature j after selecting the feature i. In this work, we propose the HUD function as the performance metric of the classifier. Since, it is the undesirability to select next feature, hence lower the value of HUD, lower the undesirability of feature j and higher the probability of selection.

Hence, HUD(i, j) = FAR (false alarm rate) of the classifier. Lower the false detection by a given set of features, lower its undesirability.

(b) An iteration's best solution is also calculated by the evaluation metric of the classifier. In the mentioned work, we determine this metric by two values (a) detection rate of the classifier, (b) number of features in the subset selected. Hence the quality function is defined as,

$$q(T\ IWD) = \frac{SL(T\ IWD)}{DR(T\ IWD)} \quad (9)$$

where $SL(T_{IWD})$ denotes the cardinality of the solution of T IWD. Cardinality refers to the size of the feature subset selected by the respective IWD and DR refers to the detection rate achieved by that IWD solution through SVM. Lesser the cardinality and greater the detection rate, lesser the value of quality function and greater the possibility of selection, since,

T IB = arg (min) q T IWD∀T IWD

The iteration-best solution selects the solution of that IWD which gives minimum value of the quality function.

The pseudo-code for the proposed IWD-ACO based feature selection is as given below in Table I.

Table I. Proposed Algorithm

HYBRID IWD+ACO Algorithm

Input: Complete Graph of Features G(N, E)
Output: Best Optimal Feature Subset(T^{TB})
1: Initialize the static parameters
2: **while** number of iteration <MaxIter **do**
3: Initialize the dynamic parameters
4: Distribute N_{IWD} number of IWDs randomly on the graph nodes
5: Update the visited node list$\mathbf{V_c}^{\mathbf{IWD}}{}_{\mathbf{k}}$ to include the source node just visited.
6: ***For*** each IWD, perform,
7: Choose the next node according to $\mathbf{p_i}^{\mathbf{IWD}}{}_{\mathbf{k}}$
8: Append the next node in the visited list.
9: Move drop IWD to the next selected node.
10: Update the values of
i) velocity $vel^{IWD}{}_k(t+1)$
ii) soil carried by IWD on path from node i to j as Δsoil(i,j)
iii) soil of the IWDsoil^{IWD}
iv) soil of the path soil(i,j)
11: ***End For***
12: ***For*** each IWD solutions
i) Calculate Subset Length: $SL(T^{IWD})$
ii) Call **ACO** Algorithm as follows-
a) Begin
b) Initialize Parameters-
α,β,ϕ,ϕ,m,τ □,ω,T
c) Let t =1
d) Distribute m ants over the

features selected by the IWD
e) while t<T
f) for each ant k=1...m
do start list S□(t)={}
g) from current node, select next j depending upon the pheromone value
h) end for
For each node i do
Apply pheromone evaporation
Update pheromone and other parameters
End for
j) t=t+1
k) end while
l) return the best solution S□(t)

13: ***End For***
14: Iteration's best solution T^{IB} is calculated As $T^{IB=}$ S□(t)
15: Update the soil value of the path followed by IWDs of iteration's best solution T^{IB}
16: If $q(T^{TB})>=q(T^{IB})$, then $T^{TB}= T^{IB}$
17: **End While**

18: Number of iterations is incremented by 1
19: Return Total Best Solution(T^{TB})

C. *Classification with Support Vector Machine (SVM)*

SVM is a popular machine learning technique which is generally used for the classification. It is ideal for non-linear classification. SVM finds a linear optimal hyper plane to maximize the margin of separation between the two classes. Suppose we have N training data samples $\{(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)\}$. Here $_{xi}\in$ Rd and $y_i\in \{+1,-1\}$. Let there be hyper plane represented by (w, b),where w refers to weight vector and b denotes bias. A new object x can be classified with the following equation

$$f(x)sin(w.x+b) = sin(\sum_{i=1}^{N} \propto_i \gamma_i(x_i, x) + b) \qquad (10)$$

The data is not linearly separable, but non-linear data can be represented as a higher dimensional space such that it becomes linearly separable in that feature space. The mapping is done through a kernel function *K*. kernel function of SVM is as follows:

$$f(x) = sin\left(\sum_{i=1}^{N} \propto_{i,} \gamma_i K(x_i, x) + b\right) \qquad (11)$$

Here K(xi , x)is the kernel function. For the proposed model SVM is normally used for binary classification. The dataset is classified into two classes, normal and attack. The SVM's classification performance for the most optimal subset of features is considered as the performance of the model.

4. EXPERIMENT USING KDD CUP'99 DATASET

KDD CUP'99 dataset was created from the data captured in DARPA'98 IDS evaluation program. Stolfo and Lee has been credited with preparing this dataset, which is now a widely used dataset for detection of anomalies in intrusion detection system[14][16]. It contains around 4 gigabytes of compressed raw (binary) tcpdump data collected over 7 weeks of network traffic. It can be processed as 5 million connection records, each having around 100 bytes. The Training dataset KDD has around 4,900,000 single connection vectors. Each of these have 41 features and are labelled as normal or an attack. The attacks are 22 in number while they are broadly classified in four classes as DoS, Probe, R2L, U2R.

D. *RESULT ANALYSIS*

The proposed work aimed at finding minimal and most optimal set of features from an input dataset such that the training time of the model is reduced as compared to other existing approaches.

The IWD+ACO method for feature selection was considerably able to reduce the training time of feature space by the classifier to a lowest minimum value of 0.97 min as shown in table II.

Table II. Comparison of Training time with previous work

Method	***Training Time(min)***
IWD+SVM[18]	1.15
DSSVM[14]	3.5
IWD+ACO(proposed)	0.97

The graphical comparison of training times can be as shown in figure 2 which depicts huge differences in training time. As can be seen from the comparison most of the earlier works on feature selection relied on exhaustive search methods like SVM, RST etc.

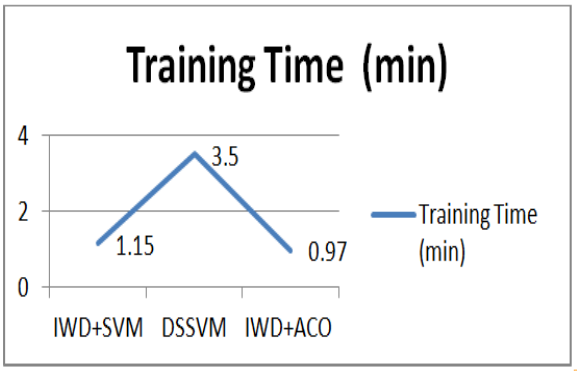

Figure 2. Graphical Comparison of Training time with previous works

Some filter methods like PCA, information gain (IG) and information gain ratio (IGR) were also used. The results obtained by the ranking methods like IG and IGR included 20 and 19 features in their dataset. While the method of lowering the dimensionality with i.e PCA method could only reduce up to 23 features. The application of Rough Set Theory on the other hand could reduce only upto 29 features. The table shows that the number of features being selected as most optimal subset got considerably reduced in researches when metaheuristic algorithms were started to be used. For eg: use of GA and GPC brought considerable reduction in number of features to 10.

While the use of nature inspired Intelligent Water Drops algorithm lowered the number to 9. On the other hand by use of ACO, when applied for different classes generated a

maximum of 8 features for Probe class and a minimum of 3 for R2L class. By combining the concepts of two computationally most efficient algorithms our proposed work (IWD+ACO) was able to bring down the number of features in the subset to 7 with a considerable reduction in training time to 0.97 min. This computationally effective result can be seen as better than the rest of the models.
The number of features as attained by the IWD-ACO method and other comparative methods are as shown in table 2.

Table 3 Comparison with other existing models

Authors	***Feature Selection Technique***	***DataSet Used***	***Feature Subset Achieved***
Acharya.N, Singh.S[18]	IWD	KDD CUP'99	9
Aghdam and Kabiri [3]	ACO	KDD CUP'99	Probe class=8 R2L=3
Aslahi-Shahri B. M et.al[10]	GA	KDD CUP'99	10
Ahmad Iftikhar et.al[19]	GPC	KDD CUP'99	10
Rung-Ching Chen et.al[8]	RST	KDD CUP'99	29
Sung and Mukkamala[6]	SVM	KDD CUP'99	13
Heba F. Eid et. al[9]	PCA	NSL-KDD	23
Safaa O. Al-mamory et.al [5]	Information Gain	KDD CUP'99	20
Sannasi Ganapathy et.al[7]	Information Gain Ratio	KDD CUP'99	19
Proposed Work	**IWD+ACO**	**KDD CUP'99**	**7**

5. CONCLUSION

The proposed work uses a combination of two metaheuristic algorithms namely Intelligent Water Drop (IWD) and Ant Colony Optimization (ACO) for optimal feature selection, where IWD has been used to generate feature subsets which are further optimized by using ACO in order to generate the minimal length feature subset which reduces the training time of classifier. The experiments were conducted using KDD CUP'99 dataset. The IWD+ACO algorithm was able to reduce the number of features from 41 to exact 7 in a computationally effective manner by reducing the training time to 0.97 min. This result was found to be better than the other existing models in terms feature selection and training time.

REFERENCES

[1] S. Singh and S. Silakari, "A survey of cyber-attack detection systems," International Journal of Computer Science and Network Security 9.5 (2009): 1-10.

[2] K. El-Khatib, "Impact of feature reduction on the efficiency of wireless intrusion detection systems," IEEE TRANSACTIONS on parallel and distributed systems 21.8 (2010): 1143-1149.

[3] M. H. Aghdam, and P. Kabiri, "Feature Selection for Intrusion Detection System Using Ant Colony Optimization," IJ Network Security 18.3 (2016): 420-432.

[4] M.A. Ambusaidi , X. He, , P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," IEEE transactions on computers, 65(10), pp.2986-2998..

[5] S. O. Al-mamory and F. S. Jassim, (2015), "On the designing of two grains levels network intrusion detection system," Karbala International Journal of Modern Science, 1(1), 15-25.

[6] S. Mukkamala and A. Sung, (2003), "Feature selection for intrusion detection with neural networks and support vector machines," Transportation Research Record: Journal of the Transportation Research Board, (1822), 33-39.

[7] S. Ganapathy, K. Kulothungan, , S. Muthurajkumar, M. Vijayalakshmi, P. Yogesh, A. Kannan, (2013), "Intelligent feature selection and classification techniques for intrusion detection in networks: a survey," EURASIP Journal on Wireless Communications and Networking, 2013(1), 271.

[8] R. C., Chen, K. F. Cheng, Y. H. Chen, , and C. F. Hsieh, (2009, April), "Using rough set and support vector machine for network intrusion detection system," In Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on (pp. 465-470). IEEE.

[9] F. E. Heba, A. Darwish, A. E. Hassanien, and A. Abraham, (2010, November), "Principle components analysis and support vector machine based intrusion detection system," In Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on (pp. 363-367). IEEE.

[10] B. M. Aslahi-Shahri, R. Rahmani, M. Chizari, , A. Maralani, M. Eslami, M. J. Golkar and A. Ebrahimi, (2016), "A hybrid method consisting of GA and SVM for intrusion detection system," Neural Computing and Applications, 27(6), 1669-1676.

[11] H. Shah-Hosseini, (2009), "Optimization with the nature-inspired intelligent water drops algorithm," In Evolutionary computation. InTech.

[12] B. O. Alijla, C. P. Lim, A. T. Khader and M. A. Al-Betar, (2013, March), "Intelligent Water Drops Algorithm for Rough Set Feature Selection," In ACIIDS (2) (pp. 356-365)..

[13] D. Martens, B. Baesens and T. Fawcett, (2011), "Editorial survey: swarm intelligence for data mining," Machine Learning, 82(1), 1-42.

[14] C. F. Tsai, Y. F. Hsu, , C. Y. Lin, and W. Y. Lin, (2009), "Intrusion detection by machine learning: A review," Expert Systems with Applications, 36(10), 11994-12000.

[15] S. Singh and S. Silakari, (2009), "A survey of cyber-attack detection systems," International Journal of Computer Science and Network Security, 9(5), 1-10.

[16] F. Iglesias and T. Zseby, (2015), "Analysis of network traffic features for anomaly detection," Machine Learning, 101(1-3), 59-84..

[17] F. Haneef, G. R. Kushwaha, and A. K. Dubey, (2011, June), "Analysis with Data Mining and Ant Colony Algorithm for Implementing of Object Pool Optimization," In Communication Systems and Network Technologies (CSNT), 2011 International Conference on (pp. 313-317). IEEE.

[18] N. Acharya, and S. Singh, (2017), "An IWD-based feature selection method for intrusion detection system," Soft Computing, 1-10.

[19] I. Ahmad, M. Hussain, A. Alghamdi and A. Alelaiwi, (2014), "Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components," Neural Computing and Applications, 24(7-8), 1671-1682.