Enhanced Hybrid Model of Support Vector-Grey Wolf Optimizer Technique to Improve the Classifier's Detection Accuracy in Designing the Efficient Intrusion Detection Model

Vidhya Sathish[1], P. Sheik Abdul Khader[2]

[1*]Research Scholar
Department of Computer Applications
B.S. Abdur Rahman Univeristy
Vandalur, Chennai – 48.

[2]Professor & Director Data Centre
Department of Computer Applications
B.S. Abdur Rahman University
Vandalur, Chennai – 48.

**Corresponding author's email : vidhyasathish83 [AT] gmail.com*

ABSTRACT---- ***Research over designing the intrusion detection systems is happening during a decade in tremendous way. The specified reason is to study the intrusion presence over network traffic. Based on this, analysis and classification of traffic pattern refined, to projects the importance of enhanced detection approach. Herewith, the methodologies deployed to better understand of 'abnormal' traces and paved the way for descriptive analysis. From the literature review of study, clarifies that hybrid computation approach entitled as fine-tuned work in analyzing intrusion trace illustrative when compared to single approach. The hazard of hybrid methodology is to be high resource computation this leads to step down in its absolute approach. The proposed research has framed to improve the support vector machine's classifier approach in designing the efficient detection model. This happens by additive support of grey wolf optimizer algorithm with the aim to improve the classifier's detection accuracy in exact classification of 'normal' and 'abnormal' instance traces from the modified KDDCUP99 intrusion dataset in minimal learning time. Experimentation of work outperform using WEKA simulator tool in WINDOWS operating system background.***

Keywords--- Grey Wolf Optimizer, Support Vector Machine, Enhanced hybrid approach, Detection accuracy

1. INTRODUCTION

The significance of intrusion traces presence over host based network traffic bloomed-up as a serious threat [1] impact in the internet industry. Diversified computational methodologies designed to broaden their approach with an objective to build the efficient intrusion detection system. These contemporary methodologies classified into two varied techniques such as single and hybrid techniques. In single technique, most of the methodologies focused with neither machine learning-based approaches[2] nor data mining based approaches. The pitfall behind these methodologies find hard to resolve and analyze the performance of every intrusion instances presence over network traffic. Therefore, progress of single technique limits within preliminary extent in analyzing the every instance trace. In other hand, to make the detection approaches tougher hybrid approaches are designed with contemporary evolutionary computational methods. Very few of experimentation had conducted on these techniques are framed with machine learning based combined approaches to show-off their detection accuracy up to maximum extent. These approaches had achieved its succeed rate in an initial level.

Intrusion Detection Systems [3] generally classified into two varied approaches observed in identification of pattern analysis and they are known to be Signature based detection approach and Anomaly based detection approach i.e., neither rule-based nor expert-system based. In Signature (rule-based) based detection approach only able to trap known signature pattern in identification of intrusion traces i.e., able to trap the intrusion trace which is already trained by system. Simultaneously, Anomaly (expert-system) based detection system is designed to trap unknown signatures identified as intrusion trace. Detection methodologies [4] are highly framed by data mining based detection approaches and machine learning based detection approaches. Herewith, methodologies are designed to extract the data for pattern analysis as a reference of 'normal' and 'abnormal' network traffic traces which represent in the form of single approach.

The drawback behind these approaches fails to analyze the presence of every categorization of instance trace over network traffic. Therefore, utilization of contemporary computational technique as hybrid model designed to study the performance analysis of intrusion traces has obtained its succeed rate in a preliminary limits due to high resource computation.

Based on this challenging task, Proposed Research has designed to strengthen Support Vector Machine classifier's detection accuracy as hybrid computation technique with Grey Wolf Optimizer algorithm. To obtain the goal to study, analyze and classify through the performance of every instance collected from modified KDDCUP99 dataset to classify accurately by normal and abnormal instances with in minimal learning time. Herewith, the four varied intrusion instances of data was taken from modified intrusion dataset known as Probe attacks, User-to-Root attacks, Root-to-Local attacks, Denial-Of-Service attacks along with 'normal' network pattern taken for evaluation process. Experimentation of work has performed using WEKA simulator tool in WINDOWS operating system background. The reason to choose the WINDOWS is that Microsoft WINDOWS most adopted for passage of viruses internally or externally for Hackers in worldwide. To overcome from this, Proposed Research takes initiative to perform the execution task in this environment.

The specified reason to project the Grey Wolf Optimizer algorithm [5] in Proposed framework is identified to be population based meta-heuristics algorithm which simulates the leadership hierarchy and haunting mechanism of live 'grey wolves' in nature i.e., objective to search, locate and offer the fittest optimal for the solution required. Grey Wolf Optimizer designed with hierarchical behavior which categorizes are 'alpha', 'beta' and 'omega'. In 'alpha' based search agent defined to be decision maker in organizing the other categories to find the location of the fittest optima. In 'beta' based search agent subordinate with the decision maker to organize the plan for providing the fittest optimal solution. The remaining search agents seem to better option to subordinate the alpha and beta search agents but 'omega' search agent will be dominated by 'delta' search agents to offer the third fittest optimal for required solution.

2. RELATED WORK

Traditional approaches like firewall security fails to identify the intrusion attack because abnormal traces of intrusions resemble the normal traffic pattern. Therefore intrusion Detection Systems, groomed with diversified mechanisms as a line of defense [6] [7]denoted as a necessity. Some of the techniques have been explained with their methodologies experimented during execution process. Utilization of data mining techniques turned as active bootstrap to start-up with diversifies intrusion detection systems. The basic strategy of experimenting data mining and machine learning based intrusion detection systems is initiates with work as every instance in the intrusion dataset specified can be labeled neither normal instance nor the intrusions type of attack. Learning algorithm will be trained over labeled dataset for Pre-processing, classification and analysis during training and testing process to build the model with an objective to study the behavior. In case of new type of intrusions had appeared, then again machine learning algorithm will be trained to study the behavior and analysis with new dataset will be labeled along with new type of intrusions. These type of detection systems effectively occupied by pattern identification based signature intrusion detection systems, whereas in variant approaches are models built based on real network traffic identification, i.e., models are designed on normal behavior in case of any other deviation occurs apart from normal behavior is identified as an intrusion. Moreover, designing the Intrusion Detection Systems able to prompt with identify the intrusion occurrences [8]. Basically two variant criteria have been associated with performance evaluation. They are known to be first, detection rate defined as ratio of correctly classified intrusions type of instances of the total number of instances in. Second, false alarm rate or also known to be false positives which is defined as the number of 'abnormal' network traffic traces has been assigned as 'normal' network traffic traces to the total number of network traffic parameters. Designing a new platform for Intrusion Detection techniques with an efficient [9], versatile approach is a challenging task.

Recent days in the Intrusion Detection Systems [10] field of research, identifies that cope with a single technique pretend to be failure model due to inefficiency in analyzing today's network security threats. So it is urged to combine the secure methodologies either within same pattern or variant combined techniques. Basic consideration of designing Intrusion Detection Systems [11] are data collection; Pre-processing; classification and clustering; intrusion response and finally ends with reporting. Criteria mentioned above undergone by standardizing two essential performance evaluation and they are known to be (i) high detection rate is defined as the ratio of correctly classified number of instances in the dataset and (ii) low false alarm is defined as the ratio of misclassified as normal behavior to the total number of instances from available dataset. Therefore herewith while constructing new platform intrusion detection system should keep in mind that able to achieve high detection accuracy with low false positive rate. Recent year's hybrid based computation techniques for constructing Intrusion Detection Systems being embraced to provide optimal solution over the high complexity of intrusions occurrence. The reason behind is more adaptive to handle complex programs without any need overlook to existence. Moreover, agents being trained over classification rules for anomaly based cluster classification and keep track of intrusions in the real network traffic search space. Some of the famous categorization techniques employ with Intrusion Detection Systems [12] based Ant Colony Optimization; Particle

Swarm Optimization. There are subcategories for every technique and they are known to be (i) Ant Colony Optimization technique, framed for induction of rules classification and for identifying the origin of attack. (ii) Particle Swarm Optimization (PSO) based intrusion detection systems focused with neural network hybrid approaches; PSO and Support Vector Machine approaches; PSO & K-means approaches; PSO for induction of classification rules. (iii) Ant Colony Clustering (ACC) based intrusion detection systems focused with ACC & Self Organizing Map (SOM) hybrid approaches and ACC and Support Vector Machine hybrid approaches. This paper also discusses the existence of computation techniques based intrusion detection techniques paved the way to carry out enhanced proposed research work. Generally, Intrusion Detection Systems has been segregated into three variant categories. They are known to be (i) detection techniques of various machine learning algorithms such as neural networks, fuzzy logic, k-means and data mining based techniques (ii) deployment of evolutionary computation in intrusion detection systems (iii) hybrid based classifier and optimization approaches based detection techniques.

The system designed for anomaly based intrusion detection[13], identification with the mixture of neural networks and decision trees, intent to identify both unknown and known intrusions. Decision trees widely used for known intrusions whereas, most of the approaches found victory in identifying the known intrusions as misuse detection techniques. Simultaneously in identification of unknown intrusions is made by mixture of supervised and unsupervised neural network. This happens by splitting up the attack classifications into smaller categories based on hybrid Self Organizing Map for a complete grouping. [14] discussed the complexity of intrusion detection dataset and also pointed out the redundancy of attributes with no more contribution to intrusion detection experimental work. Moreover, this paper progressed to work to build classifier with specification of attributes set. This can achieved its detection rate and low false alarm rate within limited resources. Some papers have discussed about data mining based intrusion detection systems for real-time environments. Proposed [15] the system for signature pattern identification to make system highly interpret with known intrusion identities. The major drawback behind these approaches is failing to identify the anomalies in network traffic. Another approach of data mining based intrusion detection system designed by [16] as modified random forest algorithm to build efficient model through experimentation of KDDCUP99 dataset. Since a single type of detection technique cannot detect all type of attack classes, most of the approaches focused with the combination of two or more techniques belong to same categorization algorithm model nor variant approaches. Utilization of genetic algorithm technique found to be tremendously leveraged to boost the detection performance in moderate level. Based on this, [17] proposed the detection technique using Genetic Algorithm with a goal of increasing the detection rate in high accuracy along with low false positives rate. But performance evaluations are limited in detecting attack classes during the training phase.

Intrusion Detection System has vast area of research due to its mission; critical nature has been attracted towards to build the model through the combination of evolutionary computation in diversifying setups. Some of the techniques have been discussed below. In Bees Colony Optimization inspired as stochastic, random-search technique. The objective of this Bees Colony Optimization is to build a multi agent system to solve the complex combinational optimization problems within computation time. Generally, this technique uses an analogy in the way in which bees searching for food and the way in which optimization algorithms search for optimum of combinational optimization problems. The critic behind these optimization techniques is parameters are needed to tune for searching process. Particle Swarm Optimization based approach embraced with victorious evolutionary computation techniques due to its easy implementation and shares the functional similarities with Genetic Algorithm. In other words, the system is being startup with a working out of random solutions and penetrating for optimize by updating generations. Particle Swarm Optimization based intrusion detection technique designed with combinational computations such as neural networks, support vector machine along rough set theory technique. Other variant classifier approaches known to be Kernel Principal Component Analysis, Radial Basis Neural Networks and Particle Swarm Optimization.

proposed model [18] by hybrid Artificial Neural Network (ANN) and PSO algorithm additionally with Rough Set Theory. Here Rough Set Theory determined as Pre-processing processes to cleanup redundancy instances or elements of ANN with intent to select specified proportional subset of input attributes and PSO is employed to optimize the parameters of ANN and thus improve the performance of ANN intrusion detection. An experimental result has achieved its detection accuracy in higher stability. Likewise, another variant approach is experimented by constructing the model using improved Particle Swarm Optimization and Support Vector Machine [improved PSO-SVM] along with Rough Set Theory for feature reduction additionally with parameter optimization. Initially Rough Set Theory is used as Pre-processing step to overcome redundancy and split up the noisy attributes and also redefine the input attribute space in right proportional way from KDDCUP99 dataset before training process starts up. Then improved Particle Swarm Optimization has been applied to the SVM classifier intent to optimize the parameters along with improving the accuracy of SVM classifier. The keen objective of this designed model is to build optimal search space from refined input attributes during training and testing processes. In deliberation of other classifier based hybrid working out are discussed here.

The technique [19] has fuzzy clustering Artificial Neural Networks combination of three classifiers known to be Decision Trees, Back Propagation Neural Networks and Naïve Bayes. Some of the well-known performance metrics

taken into account are detection rate, precision, recall, training time and f-measure. Performance metrics measured with critical aspects and they determined the Fuzzy Clustering Artificial Neural Networks proven its better performance when compared to existing classifiers taken into account for every attack class specified in the intrusion dataset. But the critics of this approach are analysis of performance metrics is very limited for every attack instance. Other variant approach experimented by [20] using the technique of decision tree, J48 and Naïve Bayes to find out high detection accuracy in intrusion identity. Performance metrics deployed with ROC curve, precision, recall, f-measure, false positives and error rate along with time taken to design the model. Though, a Naïve Bayesian approach found to be better performance when compared to other classifiers. But Decision Tree Algorithm found to be strong enough in detection of new type of intrusions.

Ant Colony Optimization [21] based Intrusion Detection technique determined as behavior-altering agents (known as pheromones). In other terms, pheromones are chemical species produce with changes the behavior of other species belongs to the same category. Generally, ants initially move randomly to locate the availability of food source. They carry food and returns to their nests with pheromone concentration have a higher probability of selects the shortest path. The Ant Nag algorithm was one of the first approaches that introduced Ant Colony Optimization in intrusion detection perceived to set all types of intrusion activities as Network Attack Graph (known as NAG). Here each edge represents critical exploit and whole graph shows from initial to destination node as an attack scenario shown. Based on the representation of graphs, ants actively iterated to cover critical exploits until the attack scenario is charged over. Though system to be highly relying on accuracy of vulnerabilities involved in attack scenario, the process of execution seems to be very tedious and complex utilization of Ant Colony Optimization (ACO) based Intrusion Detection Systems has been determined with two variant categories. They are known to be in first, deploy the ACO approach in identifying the origin of the attack takes place and in second, ACO technique used for creating set of rules to exploit the network traffic classification as normal or into the specification of attack classes. Another approach of pheromone paradigm proposed to separate Intrusion Detection Systems as independent detection units make them to communicate by increasing network traffic load without direct conversation intent to create security vulnerabilities thereby reduces the misjudgment in a concise manner. Herewith, detection rate analyzes the network behavior that is assigned to and finally produces the suspicious value. The value will be compared and if exceeded the threshold value then proceed to start alerts. Else general information is gathered and stored in the database where it is perceived as repository of pheromones.

3. OVERVIEW OF PROPOSED FRAMEWORK

The overview of Proposed Research is crafted by five major points. In first, initialize Grey Wolf Optimizer parameters for dataset classification. In second, discussed about how prediction value method is calculated for proposed research through designed parameters utilization to work. Herewith every single parameter determined to partitioned the set of modified data into ‘n’ subsets. In third, performance will be evaluated by accepting every parameter as an object from the modified dataset and returns the floating point value similar to output of prediction value method. Herewith performance metrics build to evaluate the improvised classifier’s detection accuracy proficient by ‘pairwise matrix’. In four, discussed with metrics used for experimentation and also specified about attributes listed. Finally process reach out until training the features (i=41) is achieved in an minimal learning time to build the model.

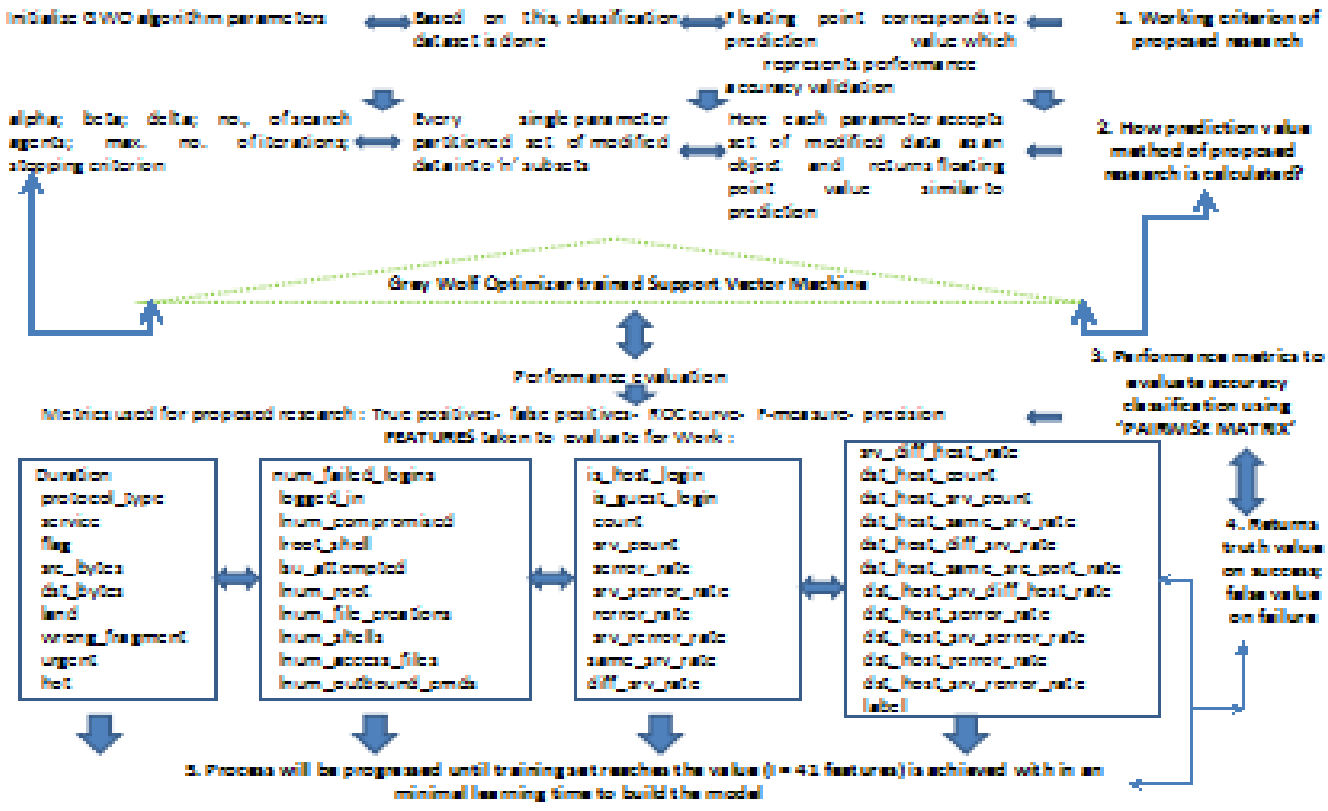

Figure 1 : Overview of Proposed Framework

From the **Figure.2** below discuss the architecture of internal flow structured with split-up process of Pre-Processing and Post-Processing experimentation of work i.e., illustrative analysis of before and after training-up the classifier with Grey Wolf Optimizer algorithm. In Pre-Processing experimentation, Support Vector Machine classifier was trained using Sequential Minimal Optimization(SMO) technique to evaluate modified set of instance classes grouped from every attack class categorization of KDDCUP99 dataset which precise as 'numeric value' to validate its correctness. Herewith, SMO predictor value method illustrate the performance captured while analyzing the output phrase of Support Vector Machine technique. Therefore, subjective of this experimentation work is to idealize the detection accuracy; false positives and learning time obtained during crafting the SVMSMO' technique.

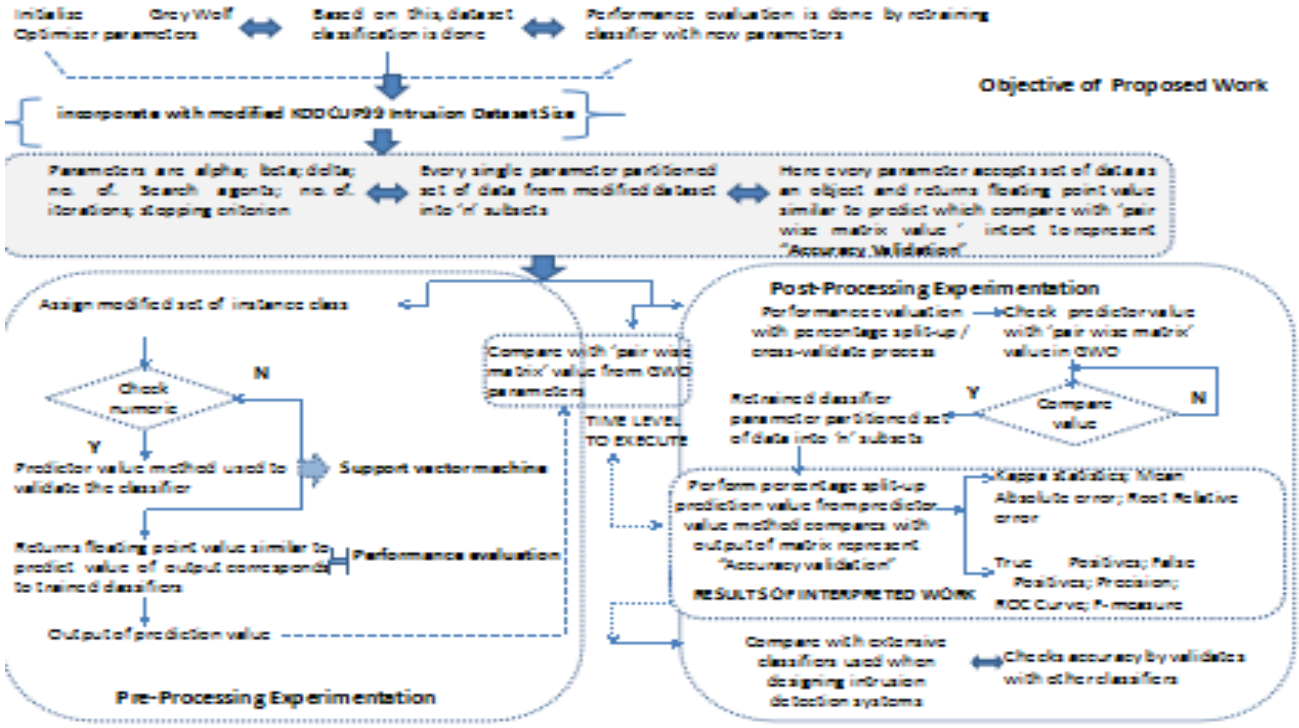

Figure 2 : Architecture of Proposed Internal Layout

In Post-Processing experimentation work, enhanced SVM approach has been validated by deploying the modified set of instances which was collected from Pre-Processing execution setup intent to show-off their result analysis in achieving the detection accuracy in classification of 'true' and 'false' positives and also time taken to build the model.

4. IMPLEMENTATION FRAMEWORK

4.1 Proposed Algorithm Steps:

Step 1: Load the Grey Wolf Optimization(GWO) stored in the file 'sample. model'

$ svm = GWO algorithm :: SVM (model => 'sample. model');

Step 2: classify a dataset

$ ds1 = GWO algorithm :: SVM : : Dataset (Label = 1, Data=> [0.12, 0.25, 0.33]);

$ res = $ svm -> predict($ds);

Step 3: Train a new SVM classifier on some modified datasets

$ svm -> train (@tset);

Step 4: Change some of the SVM parameters to calculate the fitness of each search agent

$ svm -> alpha($\mathbf{X}_\alpha$) ; # best search agent

$ svm -> beta ($\mathbf{X}_\beta$);# second best search agent

$ svm -> delta ($\mathbf{X}_\delta$); # third best search agent

Step 5: Retrain the SVM with new parameters

$svm -> retrain ((while t < max number of iterations

for each search agent

update the position of current search agent by equation

end for

Calculate the fitness of all search agents

Update $\mathbf{X}_\alpha$, $\mathbf{X}_\beta$ and $\mathbf{X}_\delta$

t = t+1

end while

return $\mathbf{X}_{\alpha}$));

Step 6: Perform cross validation on the training set

$ accuracy = $ svm -> validate (i = 1,2,....41 features), the pair wise confusion matrix in GWO algorithm is calculated as

		Predicted Value	
Actual Value		TN	FP
		FP	TP

Step 7: Repeat the steps until i =41

Step 8: Save the model to a file

$ svm -> save ('SVMGWO. model');

Step 9: Load the saved model from the file

$ svm -> load ('SVMGWO. model');

4.2 Result Discussion :

4.2.1. Performance notification during Pre-Processing experimentation:

Experimentation was conducted using two variant modes of evaluating the classifier technique. Herewith, cross-validation test and percentage-split test to outperform the output model presence is exactly the same i.e., validation will modify the only values of 'true positives' rate. Pre-Processing experimental setup split into two varied instances of attack classes collected from modified dataset. In first, '84562' instances was deployed using 'percentage-split' evaluation process i.e., by default percent split-up with 66% for training the classification period and remaining left for testing the classifier approach.

=== Run information ===

Time taken to build model: 391.63 seconds
=== Evaluation on test split ===
Time taken to test model on training split: 8.72 seconds
=== Summary ===

Table 1 : Result view of Pre-Processing experiment of '84562' instances

Scheme	weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007"	
Relation	kdd_cup_1999	
Instances	84562	
Attributes	42	
Correctly Classified Instances	28708	99.8504 %
Incorrectly Classified Instances	43	0.1496 %
Kappa statistic	0.9974	
Mean absolute error	0.0793	
Root mean squared error	0.1961	
Relative absolute error	156.998 %	
Root relative squared error	123.3707 %	
Mean rel. region size (0.95 level)	74.1539 %	
Total Number of Instances	28751	

In second, '91059' instances was deployed in two modes of evaluation model i.e., known to be 'percentage-split' mode and 'cross validation' mode. The specified reason to prove about the statements of under any mode of evaluation model 'output' is always the one built on all the data. In otherterm, cross-validation; percentage-split are just methods for getting an estimate of how well the model will perform on future data.

Result description of **PERCENTAGE-SPLIT** mode:

=== Run information ===

Time taken to build model: 5644.9 seconds
Test mode:split 66.0% train, remainder test
=== Classifier model (full training set) ===
LibSVM wrapper (= WLSVM)
=== Summary ===

Table 2 : Result view of Pre-Processing experiment of '91059' instances - %SPLIT MODE

Scheme	weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -seed 1	
Relation	kdd_cup_1999-weka.filters.supervised.instance.	
Instances	91059	
Attributes	42	
Correctly Classified Instances	30870	99.7093 %
Incorrectly Classified Instances	90	0.2907 %
Kappa statistic	0.9929	
Mean absolute error	0.0012	
Root mean squared error	0.0341	
Relative absolute error	0.7105 %	
Root relative squared error	11.9323 %	
Total Number of Instances	30960	

Table 2 : Result view of Pre-Processing experiment of '91059' instances - %SPLIT MODE

Result description of **CROSS-VALIDATION** mode:

=== Run information ===

Time taken to build model: 487.98 seconds
Test mode:10-fold cross-validation
=== Classifier model (full training set) ===
LibSVM wrapper (= WLSVM)
=== Summary ===

Table 3 : Result view of Pre-Processing experiment of '91059' instances - **CROSS VALIDATION** MODE

Scheme	weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -seed 1	
Relation	kdd_cup_1999-weka.filters.supervised.instance.	
Instances	91059	
Attributes	42	
Correctly Classified Instances	90866	99.788 %
Incorrectly Classified Instances	193	0.212 %
Kappa statistic	0.9948	
Mean absolute error	0.0008	
Root mean squared error	0.0291	
Relative absolute error	0.5177 %	
Root relative squared error	10.1762 %	
Total Number of Instances	91059	

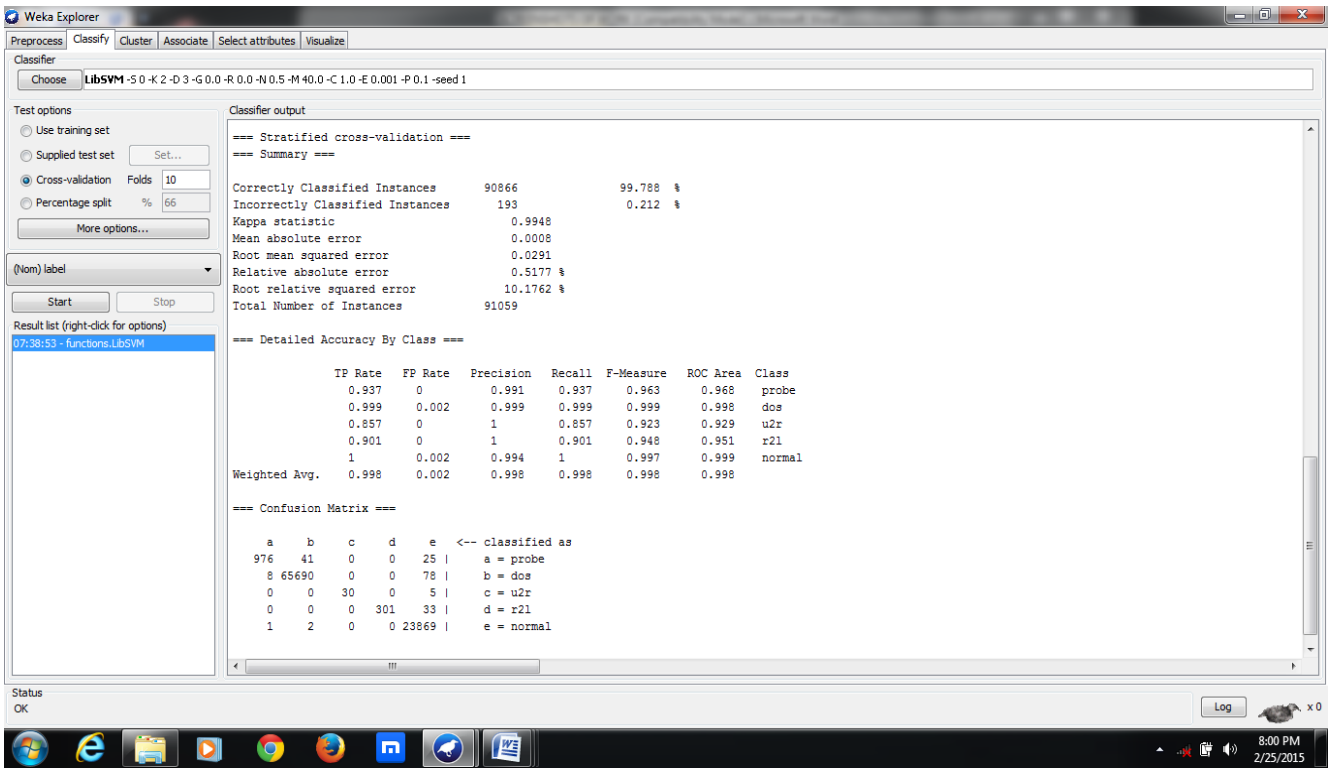

Figure 3 : Screenshot of Pre-Processing experiment of '91059' instances - **CROSS VALIDATION** MODE

From the result summary of '91059' instances exploitation was conducted using two variant modes of evaluation model projects that detection accuracy of 'Support Vector Machine' classifier approach build the interpreted work in one way. Herewith, from these execution process clarified about statement of 'any mode' of evaluation technique are just promote to estimate of how well the model will perform on future analysis.

4.2.2. Performance notification during Post-Processing experimentation:

Experimentation conducted to evaluate the performance of enhanced Support Vector Machine(SVM) classifier approach i.e., after training-up the SVM classifier with Grey Wolf Optimizer technique. Based on proven review related to testing mode of evaluation model. Herewith, two varied instances was experimented using 'cross validation' process. In first, '84562' instances of attack classification of classes along with 'normal' pattern was taken into account. Overall '41' attributes were deployed from the four main categorization of instance classes such as Probe attacks; Denial-Of-Service attacks; User-to-Root attacks; Root-to-Local attacks.

=== Run information ===

Time taken to build model: 392.49 seconds
Test mode:10-fold cross-validation
=== Stratified cross-validation ===
=== Summary ===

Table 4 : Result view of Post-Processing experiment of '84562' instances

Scheme	weka.classifiers.functions.SMOSVMGW -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007"	
Relation	kdd_cup_1999	
Instances	91059	
Attributes	42	
Correctly Classified Instances	90959	99.8902 %
Incorrectly Classified Instances	100	0.1098 %
Kappa statistic	0.9973	
Mean absolute error	0.24	
Root mean squared error	0.3163	
Relative absolute error	146.5956 %	
Root relative squared error	110.5378 %	
Coverage of cases (0.95 level)	100 %	
Mean rel. region size (0.95 level)	80.0545 %	
Total Number of Instances	91059	

In second, '91059' instances of experimentation conducted to optimize the parameters using enhanced hybrid approach.

=== Run information ===

Time taken to build model: 205.16 seconds
Test mode:10-fold cross-validation
=== Stratified cross-validation ===
=== Summary ===

Table 5 : Result view of Post-Processing experiment of '91059' instances

Scheme	weka.classifiers.functions.SMOSVMGW -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007"	
Relation	kdd_cup_1999-weka.filters.supervised.instance.	
Instances	84562	
Attributes	42	
Correctly Classified Instances	84486	99.9101 %
Incorrectly Classified Instances	76	0.0899 %
Kappa statistic	0.9985	
Mean absolute error	0.0793	
Root mean squared error	0.1961	
Relative absolute error	157.0035 %	
Root relative squared error	123.4058 %	
Coverage of cases (0.95 level)	99.9976 %	
Mean rel. region size (0.95 level)	74.9725 %	
Total Number of Instances	84562	

5. Tabular Form Of Performance Analysis For Every Class Instances

Performance analysis of Pre-Processing and Post-Processing experimentation analyzed by the kernel mode which can be expressed in terms of k(x,y)=<X,Y> intent to show-ff the output determined by linear decision boundary which expressed in terms of original attributes as well as support vectors defined for 'normal' and 'abnormal' traces. Herewith, '84562' instances taken to shown kernel evaluations obtained during Pre-Processing and Post-Processing experimentation of work.

Classifier for classes	**Pre-Processing Evaluation '84562' Instances**	**Post-Processing Evaluation '84562' Instances**
Smurf, pod	898 (76.375 % cached)	348 (78.33% cached)
Smurf, portsweep	3359(67.618% cached)	1190(66.889% cached)
Smurf, normal	123370(60.79% cached)	40394(60.208% cached)
Pod, portsweep	1210(74.585%cached)	309(71.336%cached)
Pod, normal	64809(68.405 % cached)	14641(63.58% cached)
Portsweep,normal	60699(64.311% cached)	30158(63.357% cached)
Neptune, buffer_overflow	8748(60.53% cached)	6625 (62.244% cached)
Neptune, ipsweep	39341(62.942 %cached)	10332(62.793 %cached)
Neptune, smurf	28492(62.778% cached)	18120(68.262% cached)
Neptune, pod	18716(61.705% cached)	9794(65.794% cached)
Neptune, portsweep	27449(63.669% cached)	1257(71.488% cached)
Buffer_overflow, ipsweep	5991(66.953%cached)	329(72.142% cached)
Buffer_overflow, smurf	562(71.616%cached)	257(73.34% cached)
Buffer_overflow, pod	83(68.321% cached)	46(74.15% cached)
Buffer_overflow, portsweep	2007(69.368% cached)	363(76.305% cached)
Buffer_overflow, normal	88105(59.762% cached)	29726(63.972%cached)
Ipsweep, smurf	8421(69.03% cached)	1077(73.866% cached)
Ipsweep, pod	3237(70.29% cached)	712(82.897% cached)
Ipsweep, portsweep	24142(73.748% cached)	5264(81.278% cached)
Ipsweep, normal	383375(70.188% cached)	39774(63.926% cached)
CORRECTLY CLASSIFIED INSTANCES	**99.8504%**	**99.9101%**
INCORRECTLY CLASSIFIED INSTANCES	**0.1496%**	**0.0899%**

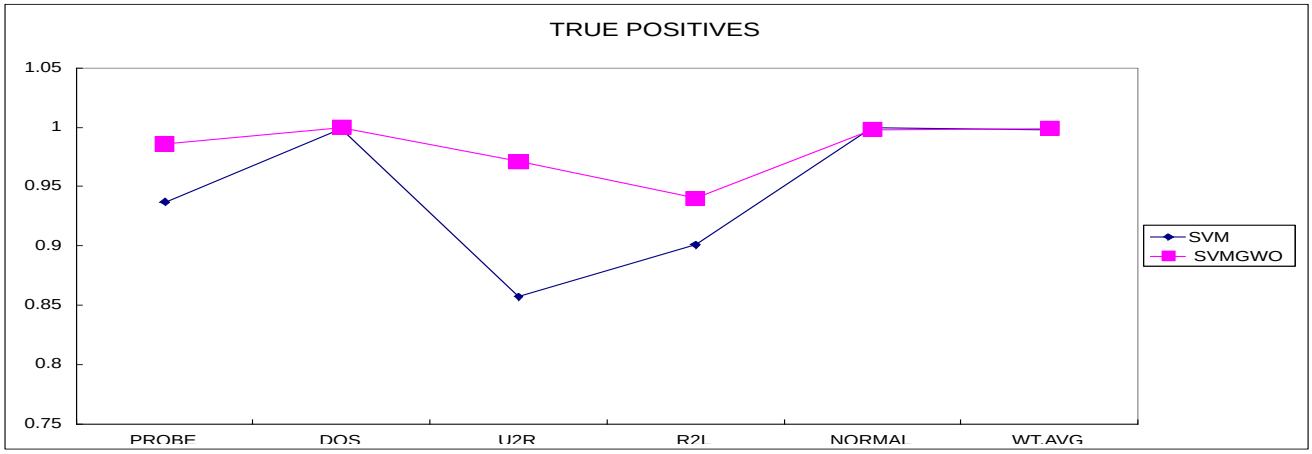

Figure 4: Comparative study on True Positives [TP] used to classify number of instances in Actual 'Positive' which were predicted to be 'Positive' in classification of 'normal' and 'abnormal' traces shown the differences between SVMGWO classifier model is accurate than Pre-Processing SVM.

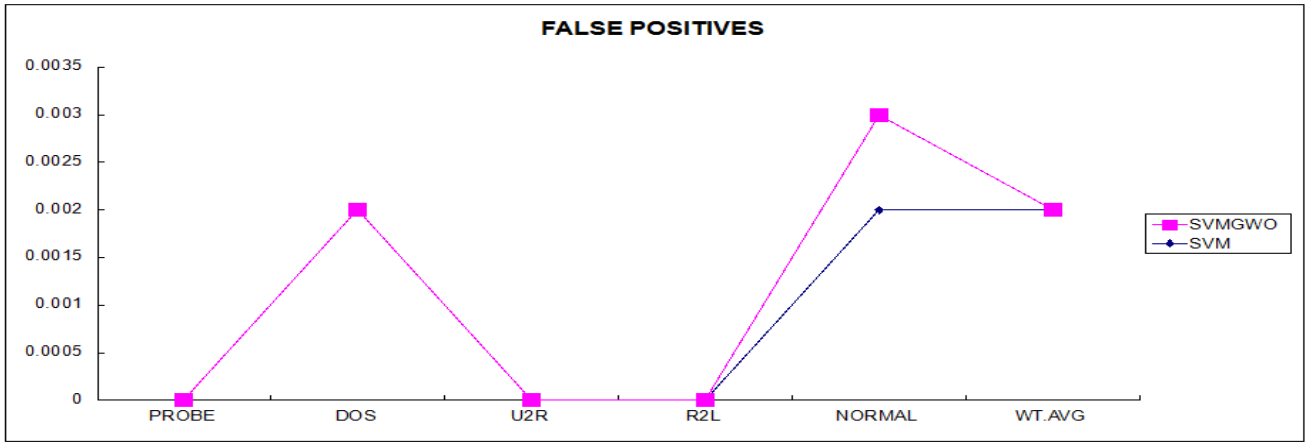

Figure 5: Comparative study on False Positives [FP] used to classify the number of instances in Actual 'negative' which were predicted to be 'positive' in classification of 'normal' and 'abnormal' traces shown the differences between proposed SVMGWO classifier model improves the effectiveness of low false positives than Pre-Processing SVM.

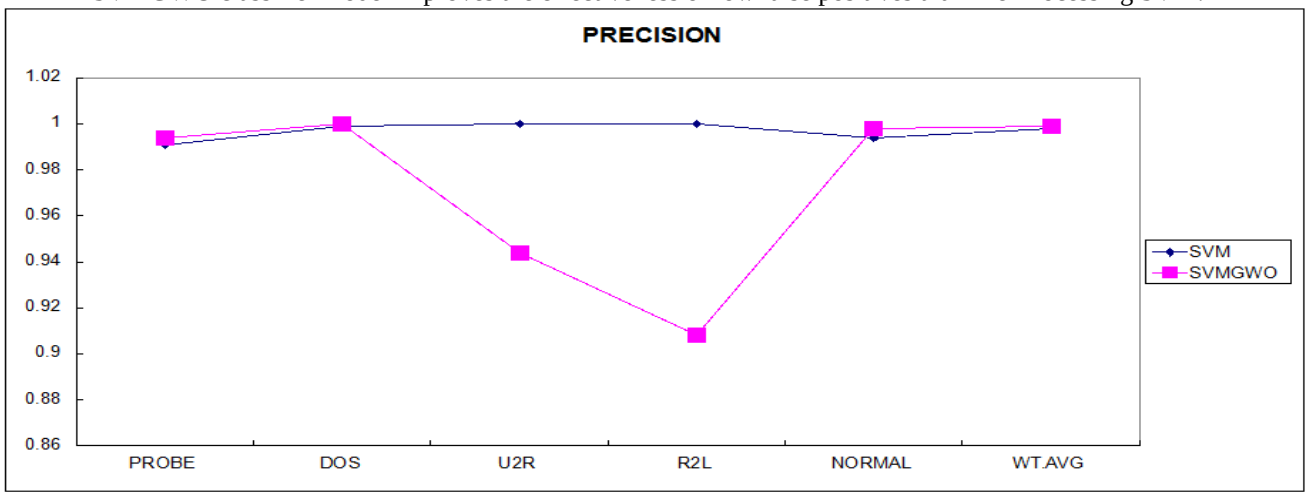

Figure 6: Comparative study on Precision in classification of 'normal' and 'abnormal' traces shown the differences between Proposed SVMGWO classifier model is found to be accurate than Pre-Processing SVM classifier model.

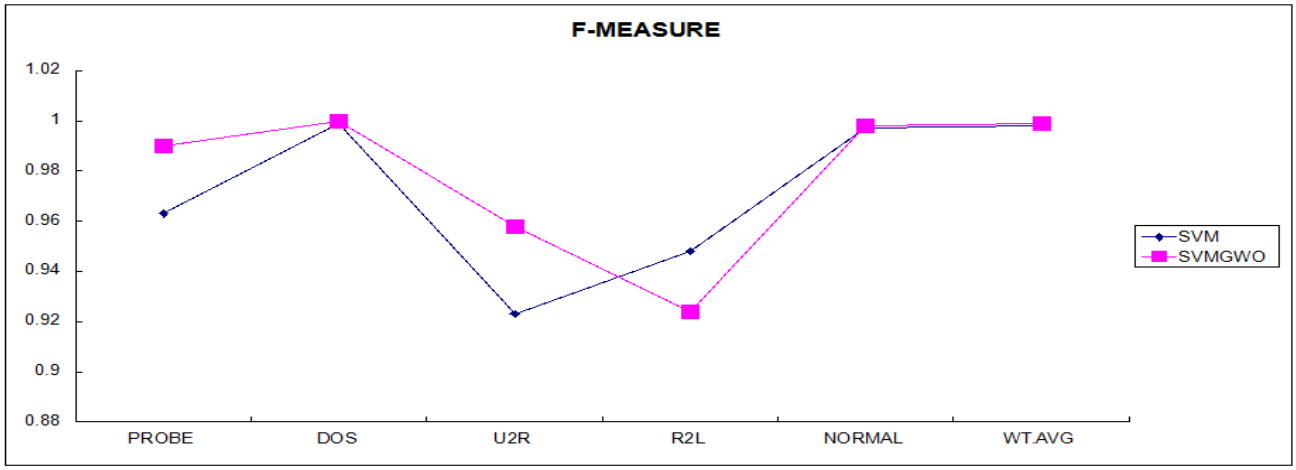

Figure 7: Comparative study on F-Measure sustain to compute the average of information retrieval from Precision and Recall metrics in classification of 'normal' and 'abnormal' traces shown the differences between Proposed SVMGWO classifier model improves accuracy rate than Pre-Processing SVM classifier model.

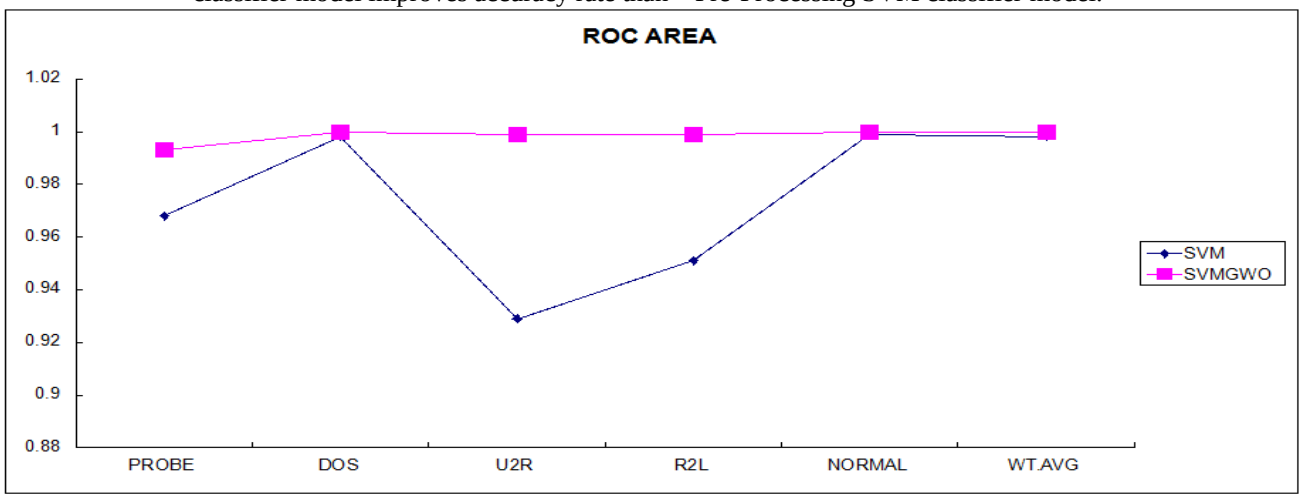

Figure 8: Comparative study on Receiver Operating Characteristic[ROC] curve for graphical display to show the trade-off between True Positives Rate and False Positives Rate in classification of 'normal' and 'abnormal' traces using Proposed SVMGWO classifier model with Pre-Processing SVM classifier model.

6. CONCLUSION

The proposed research deliberate as upgrade work in justifying the efficient intrusion detection system i.e., intend to strengthen the quality of Support Vector Machine classifier in evaluating the every categorization of instance classes individually. The performance metrics used are kappa statistics; correctly classified instances; incorrectly classified instances; precision; true positives obtained; mean rel. region size and time taken to build the model. Experimentation of work was conducted to classify the performance description of before and after training up the classifier with Grey Wolf Optimizer approach. It is concluded that proposed experimentation results is enhanced to strengthen the classifier's detection accuracy to maximum extent using pairwise matrix evaluation in justifying the accurate classification of 'normal' and 'abnormal' instances in elaborative way. It was also used for the future enhancement of extending the research in escalate the performance of other classifier approaches (extreme learning machine, relevance vector machine) with Grey Wolf Optimizer algorithm.

7. REFERENCES

[1] VidhyaSathish and P.Sheik Abdul Khader, "Deployment of Proposed Botnet Monitoring Platform using Online Malware Analysis for Distributed Environment", Indian Journal of Science and Technology, vol. 7, no. 8, pp.1087-1093, 2014.

[2] Sundusjuma, Zaitonmuda, M.A. Mohamed, and Warusia Yassin, "Machine Learning Techniques For Intrusion Detection System: A Review", Journal of Theoretical and Applied Information Technology, vol. 72, no. 3, pp. 422-429, 2015.

[3] VidhyaSathish, and P.Sheik Abdul Khader, " An Investigational Study on Intrusions based Traffic Identification using WEKA tool" , International Journal of Applied Engineering Research , vol. 10, no. 15, pp.35160-35166, 2015.

[4] J.Singh and M.J.Nene, "A Survey on Machine Learning Techniques for Intrusion Detection Systems", International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 11, pp.4349-4355, 2013.

[5] S.Mirjalili, S.M.Mirjalili and A.Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, Elsevier, vol. 69, pp.46-61, Mar 2014.

[6] P. Sapate, and S.A.Raut, "Survey on Classification techniques for Intrusion Detection", Proceedings of the Fourth International Conference on Advances in Computing & Information Technology (ACITY 2014), May 24-25, pp. 223-231, 2014.

[7] Harsimran Kaur, "Algorithm used in Intrusion Detection Systems: A Review", International Journal of Innovative Research in Computer and Communication Engineering., vol. 2, no. 5, pp.4438-4446, 2014.

[8] N.K.Sinha, G.Kumar, and K.Kumar, "A Review on Performance Comparison of Artificial Intelligence Techniques used for Intrusion Detection", International Conference on Communication, Computing & Systems (ICCCS-2014)., pp.209-214, 2014.

[9] S.K.Jonnalagadda, and I.R.P. Reddy, "A Literature Survey and Comprehensive study of Intrusion Detection", International Journal of Computer Applications, vol. 81, no. 16, pp.40-47, 2013.

[10] H.Liao, C.R.Lin, Y. Lin, and K.Tung, "Intrusion Detection System: A Comprehensive review", Journal of Network and Computer Applications, vol. 36, no. 1, pp.16-24, 2013.

[11] Prof. N.S. Chandolikar, and Prof. V.D. Nandavadekar, "Selection of Relevant Feature for Intrusion attack classification by analyzing KDDCUP99", MIT International Journal of Computer Science and Information Technology. Vol. 2, no. 2, pp. 85-90, 2012.

[12] C.Kolias, G.Kambourakis, and M.Maragoudakis, "Swarm Intelligence in Intrusion Detection: A Survey", Computers & Security, vol . 30 no. 8, pp.625-642, 2011.

[13] M.Bahrololum, E. Salahi, and M. Khaleghi, "An Improved Intrusion Detection Technique based on two strategies using Decision Tree and Neural Network", Journal of Convergence Information Technology, vol.4, no. 4, pp.96-101, 2009.

[14] D.Farid, J.Darmont, N.Harbi, N.H.Hoa, and M.Z.Rahman, "Adaptive Network Intrusion Detection Learning: Attribute Selection and Classification", Proceedings of the International Conference on Computer Systems Engineering (ICCSE 09), vol. 60, pp.154-158, 2009.

[15] N.Ye, and X.Li, "A Scalable Clustering Technique for Intrusion Signature Recognition", Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, pp.1-4, 2001.

[16] J. Zhang, and M. Zulkernine, "Anomaly based Network Intrusion Detection with Unsupervised Outlier Detection", IEEE International Conference on Communications, pp.2388-2393, 2006.

[17] G.Stein, B.Chen, A.S.Wu, and K.A.Hua, "Decision tree classifier for Network Intrusion Detection with Genetic Algorithm based feature selection", In: Proceedings of the 43rd annual south east regional conference ACM. 2, pp.136-141, 2005.

[18] W. Tian, and J. Liu, "Network Intrusion analysis with Neural Network and Particle Swarm Optimization algorithm", In: 2010 Chinese IEEE Control and Decision Conference (CCDC), pp.1749-1752, 2010.

[19] G.Wang, J.Hao, J. Ma, and LHuang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering", Expert Systems with Applications. vol.37, no.9, pp. 6225-6232, 2010.

[20] M.Panda, and M.RPatra, "A Comparative Study of datamining algorithms for network intrusion detection", First International Conference on Emerging Trends in Engineering and Technology, pp.504-507, 2008.

[21] M.Abadi, and S. Jalali, "An ant colony optimization algorithm for network vulnerability analysis", Iranian Journal for Electrical and Electronic Engineering, vol.2, no.3, pp.106-120, 2006.