American-Eurasian Journal of Scientific Research 12 (3): 154-160, 2017
ISSN 1818-6785

DOI: 10.5829/idosi.aejsr.2017.154.160

Optimal Knowledge Extraction System Based Onigapso

[1]S. Jagadeesh Soundappan and [2]R. Sugumar

[1]Department of CSE, St. Peter's University, Avadi, Chennai-600 054, India
[2]Department of CSE, Velammal Institute of Technology, Panchetti, Chennai-601204, India

Abstract: Knowledge Discovery in Databases is a process of finding useful information and patterns in data. Research in data mining continues growing in business and in learning organization over coming decades. Use of algorithms to extract the information and patterns is derived by the KDD process.This paper explores about the mining of data and finding essential information from the complex data.Initially, the databases are divided into training and testing with the dataset respectively.We are extracting the significant rules by using the combination of evolutionary algorithm and swarm based algorithm,afterextracting optimal knowledge from the dataset via rules, the data will beclassified using KNN.

Key word: Knowledge Data Discovery · Evolutionary Algorithm · Genetic Algorithm · Particle Swarm Optimization · K nearest Neighbour

INTRODUCTION

Evolutionary algorithms are randomized search procedures inspired by the mechanics of genetics and natural selection. EAs are often used as optimization algorithms andWilson, S.W., M.D. Vose and A.H. Wright, [1,2]. EAswork on a population of individuals that represent possible solutions to a problem in their chromosomes. Each individual can be as simple as a string of zeroes and ones, or as complex as a computer program. The initial population of individuals may be created entirely at random, or some knowledge about previously known solutions may be used to seed the population. The individuals with better performance are selected to serve as parents of the next generation. Evolutionary algorithms are controlled by several inputs, such as the size of the population and the rates that control how often mutation and crossover are used.

L.B. Goldberg D.E., Holland, J. Hand, D. Thierens, J. Suykens, J. Vanderwalle and B.D. Moor, [3,4] Genetic algorithms use chromosomes composed of zeroes and ones, but other encodings may be more natural to the problem and may facilitate the search for good solutions. Genetic programming encodes solutions as computer programs. The primary mechanism in GAs to create new individuals is crossover. In its simplest form, crossover randomly chooses two individuals from the pool that were selected to be parents and exchanges segments of their two chromosomes around a single randomly-chosen point. The result is two new individuals, each with a segment of chromosome from each parent. Other variants of crossover exchange material around more than one point and some researchers have experimented with recombining chromosomes from more than two parents. Some of the new solutions will be more fit than the parents, but others will be less fit.

Kennedy, J. and R. Eberhart, [5] Particle Swarm Optimization simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.

Related Work: Wen-Jun Zhang, Xiao-Feng Xie [6], A hybrid particle swarm with differential evolution operator, termed DEPSO, which provide thebell-shaped

Corresponding Author: S. Jagadeesh Soundappan, Department of CSE, St. Peter's University, Avadi, Chennai-600 054, India

mutations with consensus on the populationdiversity along with the evolution, while keeps the self-organizedparticle swarm dynamics, is proposed. Thenit is applied to a set of benchmark functions and theexperimental results illustrate its efficiency. Deb, K [7] this paper, GA's population-based approach and to make pair-wise comparison in tournament selection operator are exploited to devise a penalty function.Careful comparisons among feasible and infeasible solutions are made so as to provide a search direction for the feasible region. Once sufficient feasible solutions are found with a controlled mutation operatorthen this allows a real-parameter GA's crossover operator to continuously find better feasible solutions, gradually leading the search near the true optimum solution. Storn, R. and K. Price, [8]. This proposed, A new heuristic approach for minimizing possiblynonlinear and non-differentiable continuous spacefunctions is presented. By means of an extensive test bed it is demonstrated that the new methodconverges faster and with more certainty than manyother acclaimed global optimization methods. The newmethod requires few control variables, is robust, easyto use and lends itself very well to parallelcomputation. Cristian T.I. [9] This presents the particle swarm optimization algorithm is analyzed using standard results from the dynamic system theory. Graphical parameter selection guidelines are derived. The exploration–exploitation trade off is discussed and illustrated. Runarsson, T.P. and X. Yao, [10]. This paper introduces a novel approach to balance objective and penalty functions stochastically, i.e., stochastic ranking and presents a new view on penalty functionmethods in terms of the dominance of penalty and objective functions. Some of the pitfalls of naive penalty methods are discussed in these terms. The new ranking method is tested using a (μ, λ) evolution strategy on 13 benchmark problems.

Problem Statement: In the Existing, The evolutionary algorithms can be very time consuming. In the tremendous computational demand of fitness evaluations in the use of genetic programming for image processing has prevented researchers from doing an extensive study of the behavior of these algorithms in solving real problems.

In the Existing, Weobserve that the evolution of an image processing operator typically takes several days to complete on a single PC, making it difficult to use their algorithm in an adaptive vision system that adapts to changing environmental conditions.

Proposed Methodology: In order to extraction the useful and meaningful with optimal knowledge from the KDD it is a big challenge in the real life scenario. As we need to overcome these challenges, we introduced the combination of evolutionary algorithm with the swarm based algorithm. In this paper Genetic algorithm were considered for the evolutionary algorithm and for the swarm algorithm, Particle Swarm Algorithm was chosen in it. This combined algorithm we were able to extract the optimal rules and then these rules were fed into the classifier of KNN. The classified output of attacks gives us the accurate outcomes. We consider the performance measures of sensitivity, specificity and accuracy for the proposed method. The observation of our proposed work is better than other existing works for the attack classification.

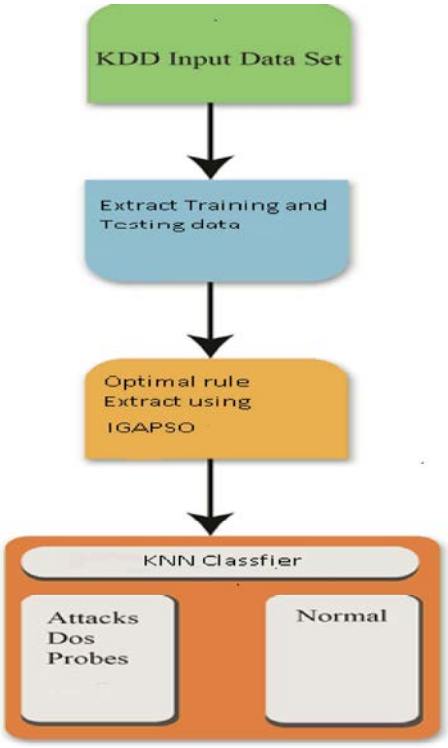


Fig. 1: Block Diagram for Proposed System

KDD Dataset: Knowledge Discovery in Databases (KDD) is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data Knowledge Discovery in Databases (KDD) is an automatic, exploratory analysis and modelling of large data repositories. KDD is the organized process of identifying valid, novel, useful and understandable

patterns from large and complex data sets. Data Mining (DM) is the core of the KDD process, involving the inferring of algorithms that explore the data, develop the model and discover previously unknown patterns. The model is used for understanding phenomena from the data, analysis and prediction. The accessibility and abundance of data today makes knowledge discovery and Data Mining a matter of considerable importance and necessity. Given the recent growth of the field, it is not surprising that a wide variety of methods is now available to the researchers and practitioners. No one method is superior to others for all cases. The handbook of Data Mining and Knowledge Discovery from Data aims to organize all significant methods developed in the field into a coherent and unified catalogue; presents performance evaluation approaches and techniques; and explains with cases and software tools the use of the different methods. In our work the KDD dataset contain 2000 for training and 1000 for testing. In that dataset we are extracted two attacks and one normal, the attacks are DOS and Probes.

Denial of Service Attacks: Denial of service (DOS) attack is an attack where the attacker creates a few calculations or memory resource completely engaged or out of stock to handle authentic requirements or reject justifiable users the right to utilise a machine. In this category, the attacker makes some computing or memory resources too busy or too full to handle legitimate request or deny legitimate users access to machine. DOS contains the attacks: 'Neptune', 'back', 'Smurf", 'pod', 'land' and 'teardrop'.

Probing Attack (PROBE): Probing is a collection of attacks where an attacker scrutinizes a network to gather information or to conclude prominent vulnerabilities. In this category the attacker attempt to gather information about network of computers for the apparent purpose of circumventing its security. Probe contains the attacks: 'port sweep', 'Satan', 'Nmap' and 'Ip sweep'. From the dataset we are taken 2000 for training and 1000 for testing then that data's are given in to the decision tree to extract knowledge, they are given below.

Random Link Attack (RLA): In an RLA, the malicious user creates a set of false identities and uses them to communicate with a large, random set of innocent users. Attackers create some fake nodes and randomly connect to regular nodes. Fake nodes form some inner structure among themselves to evade detection. From the dataset we are taken 2000 for training and 1000 for testing then that data's are given in to the decision tree to extract knowledge, they are given below.

Genetic Algorithm: Genetic algorithms are solution to a problem solved by genetic algorithms uses an evolutionary process (it is evolved).Algorithm begins with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are then selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce. This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

Step 1: Generate random population of n chromosomes (suitable solutions for the problem)

Step 2: Fitness: Evaluate the fitness f(x) of each chromosome x in the population

Step 3: New population: Create a new population by repeating following steps until the new population is complete

i). Selection: Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)

ii). Crossover: with a crossover probability cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.

iii). Mutation: with a mutation probability mutate new offspring at each locus (position in chromosome).

iv). Accepting: Place new offspring in the new population

Step 4: Replace: Use new generated population for a further run of the algorithm

Step 5: Test: If the end condition is satisfied, stop and return the best solution in current population

Step 6: Go to Step 2

PSO: PSO is initialized with a group of random particles (solutions) and then searches for optima by updating

generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is calledlbest.After finding the two best values, the particle updates its velocity and positions with following equation (1) and (2).

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) (1)

present[] = persent[] + v[] (2)

v[] is the particle velocity, persent[] is the current particle (solution). pbest [] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. usually c1 = c2 = 2.

Step 1: Initialize for each particle

Step 2: For each particle calculate fitness value If the fitness value is better than the best fitness value (pBest) in history set current value as the new pBest

Step 3: Choose the particle with the best fitness value of all the particles as the gBestfor each particle calculate particle velocity according equation (1)Update particle position according equation (2) While maximum iterations or minimum error criteria is not attained.

IGAPSO: This approach executes the two systems simultaneously and selects P individualsfrom each system for exchanging after the designated N iterations. The individualwith larger fitness has more opportunities of being selected. The main steps of the this approach are depicted below

- Initialize GA and PSO subsystems.
- Execute GA and PSO simultaneously.
- Memorize the best solution as the final solution and stop if the best individual inone of the two subsystems satisfies the termination criterion.
- Perform this process if generations could be divided exactly by the designatednumber of iterations N. Select P individuals from both sub-systems randomlyaccording to their fitness and exchange. Go to step 3

K Nearest Neighbors: KNN is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition as a non-parametric technique.

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

RESULTS AND DISCUSSIONS

KDD Dataset: This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between ``bad" connections, called intrusions or attacks and “good" normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

Evaluation Metrics: An evaluation metric is used to evaluate the effectiveness of the proposed system. It consists of a set of measures that follow a common underlying evaluation methodology some of the metrics that we have chosen for our evaluation purpose are True Positive, True Negative, False Positive and False Negative, Specificity, Sensitivity, Accuracy, F measure.

Sensitivity: The measure of the sensitivity is the proportion of actual positives which are accurately recognized. It relates to the capacity of test to recognize positive results.Where TP stands for True Positive and FN stands for False Negative

$$\text{Sensitivity} = \frac{TP}{(TP + FN)} \quad (3)$$

Specificity: The measure of the specificity is the extent of negatives which are properly recognized. It relates to the capacity of test to recognize negative results.Where TN stands for True Negative and FP stands for False Positive

$$\text{Specificity} = \frac{TN}{(TN + FP)} \quad (4)$$

Accuracy: Accuracy of the proposed method is the ratio of the total number of TP and TN to the total number of data.

$$\text{Accuracy} = \frac{TN + TP}{(TN + TP + FN + FP)} \quad (5)$$

Experimental Outcome	Condition as determined by the Standard of Truth	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Performance Analysis: The performance of the proposed knowledge extraction in attack and normal prediction methods evaluated by the three metrics Sensitivity, Specificity and Accuracy. The results of proposed work help to analyze the efficiency of the prediction process. The subsequent table II tabulates the results. Here, only the results of dataset given in table II.

Table 1: Results of the proposed Optimal Knowledge Extraction System

	TP	TN	FP	FN	Accuracy	Sensitivity	Specificity
DOS	3210	33778	218	75	0.8976	0.87543	0.88432
Probes	3477	2984	445	567	0.8798	0.86540	0.7789

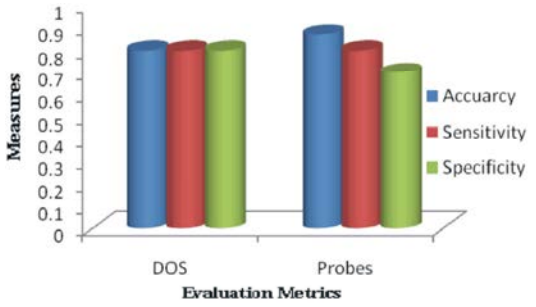


Fig. 3: Graph for results with the Performance measures Specificity and Sensitivity, Accuracy

From table 1, the evaluation metrics are analyzed for the dataset, by which we can observe the efficiency of proposed detection system. The results of the measures Sensitivity, Specificity, Accuracy are graphically represented in fig. 3. The sensitivity of two attacks is explained DOS and Probes 0.87543 and 0.86540 With these metrics, the specificity and accuracy are the main measures for evaluating the detection accuracy of our proposed system. The values of specificity for two attacks are DOS and Probes 0.88432 and 0.7789 and the values of accuracy is 0.8976and0.8798 The results get high accuracy results on behalf of the reduced error rates in the proposed system. From the fig. 3 also, we find out the minimal value of error rates for the three dataset.

Comparative Analysis: The literature review works are compared in this section with the proposed work to show that our proposed work is better than the state-of-art works. We can establish that our proposed work helps to attain very good accuracy for the attack prediction of database using KNN classifier. And also we can establish this prediction accuracy outcome by comparing other classifiers. We have utilized GA and PSO search for our Comparison in our work. The Comparison outcomes are presented in the following table 2.

Table 2: Comparison of Accuracy values in proposed vs existing method

Trial	Proposed Accuracy values	GA Accuracy values	Existing PSO Accuracy Values
1	0.962827	0.79934	0.614663
2	0.97413	0.839709	0.799813
3	0.99686	0.868677	0.817072

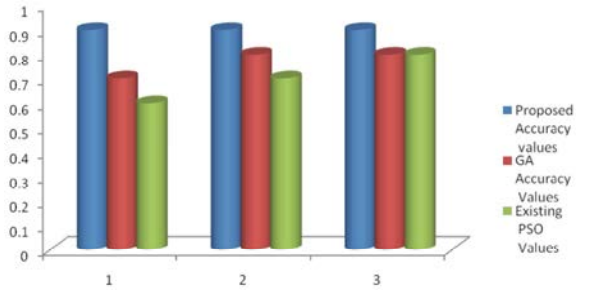


Fig. 4: Comparison for proposed Method Vs Existing Method for Accuracy

The accuracy for the Genetic Algorithmis 0.79934,0.839709 and 0.868677which is low in compared with our classifier, IGAPSO for our dataset are 0.962827,0.97413 and 0.99686.We have also compared with our classifier in PSO it will also shows a lower result which is 0.614663,0.799813 and 0.817072.

Table 3: Comparison of Sensitivity values in proposed vs existing method

Trial	Proposed Sensitivity values	GA Sensitivity values	Existing PSO Sensitivity Values
1	0.99235	0.85577	0.848291
2	0.9554	0.922	0.924021
3	0.99322	0.82347	0.968579

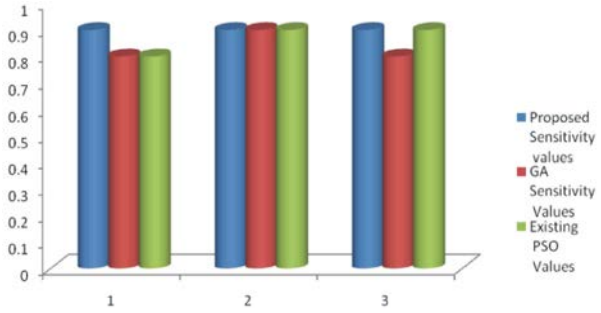
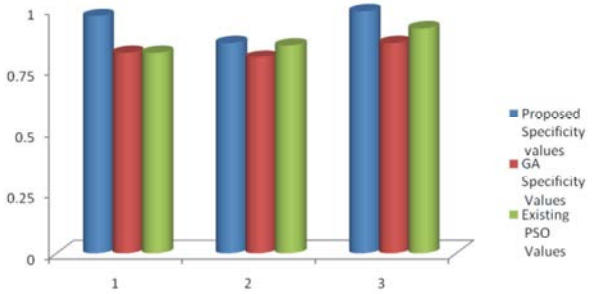


Fig. 5: Comparison for proposed Method Vs Existing Method for Sensitivity

The sensitivity for the GA is0.85577, 0.922 and 0.82347which is low in compared with our classifier, IGAPSO for our dataset are 0.99235, 0.9554 and 0.99322. We have also compared with our classifier in PSO it will also shows a lower result which is 0.848291, 0.924021 and 0.968579.

Table 3: Comparison of Specificity values in proposed vs existing method

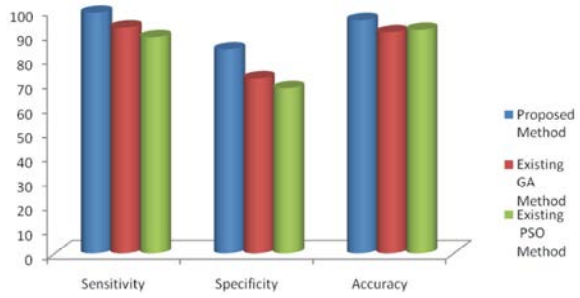
Trial	Proposed Specificity values	GASpecificity values	Existing PSO Specificity Values
1	0.97542	0.82974	0.82356
2	0.86542	0.8023	0.85320
3	0.99844	0.86423	0.9214



The specificity for the GAis 0.82974, 0.8023 and 0.86423which is low in compared with our classifier, IGAPSO for our dataset are 0.97542, 0.86542 and 0.99844. We have also compared with our classifier in PSO it will also shows a lower result which is 0.82356, 0.85320 and 0.9214.

Table 4: Comparison of Proposed Method Vs Existing Method

Metrics	Proposed Method	Existing GA Method	Existing PSO Method
Sensitivity	99	93	89
Specificity	84	72	68
Accuracy	96	91	92



The improved good accuracy outcomes of attack classification are presented by our proposed work. In comparison with the GA and PSO gives very less accuracy values for the evaluation measures. The sensitivity values of GA gives 93% and PSO is 89% but our proposed IGAPSO gives 99%. The specificity values of existing GA gives 72% and PSO is 68% but our proposed IGAPSOmethod gives 84%. The accuracy values of existing GA gives 91% and PSO gives 92% but our proposed IGAPSO gives 92%. From these outcomes, it is known that by means of KNN classifier in our work provides very good for the classify the attack and normal as it gives improved accuracy outcomes.

CONCLUSION

This paper has different stages to acquire the optimal knowledge in terms of extraction and classification. Initially the knowledge is extracted in the combination of the evolutionary algorithm with the swarm based algorithm. The proposed synthesizes the merits in both GA and PSO. Finally the attacks are successfully classified by using KNN algorithm. It is a simple and yet effective model to handle different kinds of continuous optimization problems.The performance measures of sensitivity, specificity, accuracy, were evaluated for our proposed method. Thus, we can observe that our proposed work is better than other existing works for the attack classification.

REFERENCES

1. Wilson, S.W., 1994. ZCS: A zeroth level classifier system. Evolutionary Computation, 2(1): 1-18.
2. Vose, M.D. and A.H. Wright, 1995. Simple genetic algorithms with linear fitness. Evolutionary Computation, 2: 347-368.
3. Goldberg, L.B., D.E., Holl and J.H. 1989. Classifier systems and genetic algorithms, Artificial Intelligence, 40: 235-282.

4. Thierens, D., J. Suykens, J. Vanderwalle and B.D. Moor, 1991. Genetic weight optimization of a feedforward neural network controller. In Proceedings of the Second International Conference on Artificial Neural Networks and Genetic Algorithms, pp: 658-663.
5. Kennedy, J. and R. Eberhart, 1995. Particle Swarm Optimization, from Proc. IEEE Int'l. Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948.
6. Wen-Jun Zhang and Xiao-Feng Xie, 2003. DEPSO: Hybrid Particle Swarm with Differential Evolution Operator" IEEE International Conference on Systems, Man & Cybernetics (SMCC), Washington D C, USA pp: 3816-3821.
7. Deb, K., 2000. An Efficient Constraint Handling Method for Genetic Algorithm. Computer Methods in Applied Mechanics and Engineering, 186(2-4): 311-338
8. Storn, R. and K. Price, 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optimization, 11: 341-359.
9. Cristian, T.I., 2003. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters, 85(6): 317-325.
10. Runarsson, T.P. and X. Yao, 2000. Stochastic ranking for constrained evolutionary optimization. IEEE Trans. on Evolutionary Computation, 4(3): 284-294.