ISSN: 2366-1313

A BI-OBJECTIVE HYPER-HERURISTIC SUPPORT VECTOR MACHINE (SVM) FOR BIG DATA CYBER SECURITY

[1]P.V. Rama Gopal Rao, [2]B. Vishweshwar, [3]K. Rahul Raj, [4]K. Sindhu

[1]Assistant Professor, Dept.of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad,

ramagopal.cse@tkrec.ac.in

[2]BTech student, Dept.of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad,
bejjarapukashivishweshwar@gmail.com

[3]BTech student, Dept.of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad,
rahulnani9192@gmail.com

[4]BTech student, Dept.of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad,
sindhu.kotte23@gmail.com

Abstract: *Cyber security in the content of big data is known to be a critical problem and presents a great challenge to the research community. Machine learning algorithms has been suggested as candidates for handling big data security problems. Among these algorithms support vector machines (SVMs) have achieved remarkable success on various classification problems. We formulate the SVM configuration in advance which is a challenging task that requires expert knowledge and large amount of manual effort of a trial and error. We propose a novel hyper-heuristic framework for bi-objective optimization that is independent of the problem domain. The proposed hyper-heuristic framework consists of a high-level strategy and low-level heuristics. The high-level strategy uses the search performance to control the selection of which low-level heuristic should be used to generate a new SVM configuration. The low-level heuristics each use different rules to effectively explore the SVM configuration search space. The effectiveness of the proposed framework has been evaluated on two cyber security problems: Microsoft malware big data classification and anomaly intrusion detection. The obtained results demonstrate that the proposed framework is very effective, if not superior, compared with its counterparts and other algorithms.*

Keywords: *Hyper-heuristics, big data, cyber security, optimisation*

I. INTRODUCTION

The rapid advancements in technologies and networking's such as mobile, social and Internet of Things create massive amounts of digital information. In this context, the term big data has been emerged to describe this massive amount of digital information. Big data refers to large and complex datasets containing both structured and unstructured data generated on a daily basis, and need to be analyzed in short periods of time. The term big data is different from the big database, where big data N. R. Sabar is with the Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC, Australia. Email: n.sabar@latrobe.edu.au X. Yi and A. Song are with the School of Computer Science and Information Technology, RMIT University, Australia. Email: {Xinyi, Andy. Song}@rmit.edu.au indicates the data is too big, too fast, or too hard for existing tools to handle. Big data is commonly described by three characteristics: volume, variety and velocity (aka 3Vs). The 3Vs define properties or dimensions of data where volume refers to an extreme size of data, variety indicates the data was generated from divers' sources and velocity refers to the speed of data creation, streaming and aggregation. The complexity and challenge of big data are mainly due to the expansion of all three characteristics (3Vs)- rather than just the volume alone. Learning from big data allows researchers, analysts, and organizations users to make better and faster decisions to enhance their operations and quality of life. Given its practical applications and challenges, this field has attracted the attention of researchers and practitioners from various communities, including academia, industry and government agencies. However, big data created a new issue related not only to the 3Vs characteristics, but also to data security. It has been indicated that big data does not only increase the scale of the challenges related to security, but also create new and different cyber-security threats that need to be addressed in an effective and intelligent ways. Indeed, security is known as the prime concern for any organization when learning from big data. Examples of big data cyber-security challenges are malwares detection, authentications and steganalysis. Among these challenges, malware detection is the most critical challenge in big data cyber-security. The term malware (short for malicious software) refers to various malicious computer programs such as ransomwares, viruses and scareware's that can infect computers and release important information.

via networks, email, or websites. Researchers and organizations acknowledged the issues that can be caused by this dangerous software (malicious computer programs) and therefore new methods should be developed to prevent them. Yet, even though malware is a crucial issue in big data, very little researches have been done in this area. Examples of malware detection methods include signature-based detection methods, behaviors monitoring detection methods and patterns-based detection methods. However, most of existing malware detection methods are mainly proposed to deal with small-scale datasets and unable to handle big data within a moderate amount of time. In addition, these methods can be easily evaded by attackers, very costly to maintain and they have very law success rates. To address the above issues, machine learning (ML) algorithms have been proposed for classifying unknown patterns and malicious software. ML have showing promising results to classify and identify unknown malware software. Support vector machines (SVMs) are among the most popular ML algorithms and have shown remarkable success in various real-world applications. The popularity of SVMs is due to their strong performance and scalability. However, despite these advantages, the performance of an SVM is strongly affected by its selected configuration. A typical SVM configuration includes the selection of the soft margin parameter (or penalty) and the kernel type as well as its parameters. In the literature, various methodologies have been developed for selecting SVM configurations. These methodologies can be classified based on the formulation of the SVM configuration problem and the optimization method used. An SVM configuration formulation can rely on either a single criterion, in which case k-fold cross-validation is used to assess the performance of the generated configuration, or multiple criteria, in which case more than one criterion must be used to evaluate the generated configuration, such as the model accuracy and model complexity. The available optimization methods include grid search methods, gradient-based methods and meta-heuristic methods. Grid search methods are easy to implement and have shown good results. However, they are computationally expensive, which limits their applicability to big data problems. Gradient-based methods are very efficient, but their main shortcomings are that they require the objective function to be differentiable and that they strongly depend on the initial point. Meta-heuristic methods have been suggested to overcome the drawbacks of grid search methods and gradient-based methods. However, the performance of a meta-heuristic method strongly depends on the selected parameters and operators, the selection of which is known to be a very difficult and time-consuming process. In addition, only one kernel is used in most works, and the search is performed over the parameter space of that kernel. This work presents a novel bi-objective hyper-heuristic framework for SVM configuration optimization. Hyper heuristics are more effective than other methods because they are independent of the particular task at hand and can often obtain highly competitive configurations. Our proposed hyper-heuristic framework integrates several key components that differentiate it from existing works to find an effective SVM configuration for big data cyber security. First, the framework considers a bi-objective formulation of the SVM configuration problem, in which the accuracy and model complexity are treated as two conflicting objectives. Second, the framework controls the selection of both the kernel type and kernel parameters as well as the soft margin parameter. Third, the hyper-heuristic framework combines the strengths of decomposition- and Pareto-based approaches in an adaptive manner to find an approximate Pareto set of SVM configurations. The performance of the proposed framework is validated and compared with that of state-of-the-art algorithms on two cyber security problems: Microsoft malware big data classification and anomaly intrusion detection. The empirical results fully demonstrate the effectiveness of the proposed framework on both problems. The remainder of this paper is organized as follows.

II. LITERATURE SURVEY

Novel feature extraction, selection and fusion for effective malware family classification

Modern malware is designed with mutation characteristics, namely polymorphism and metamorphism, which causes an enormous growth in the number of variants of malware samples. Categorization of malware samples on the basis of their behaviors is essential for the computer security community, because they receive huge number of malwares every day, and the signature extraction process is usually based on malicious parts characterizing malware families. Microsoft released a malware classification challenge in 2015 with a huge dataset of near 0.5 terabytes of data, containing more than 20K malware samples. The analysis of this dataset inspired the development of a novel paradigm that is effective in categorizing malware variants into their actual family groups. This paradigm is presented and discussed in the present paper, where emphasis has been given to the phases related to the extraction, and selection of a set of novel features for the effective representation of malware samples. Features can be grouped according to different characteristics of malware behavior, and their fusion is performed according to a per-class weighting paradigm. The proposed method achieved a very high accuracy (≈ 0.998) on the Microsoft Malware Challenge dataset.

Efficient string matching: an aid to bibliographic search. Communications of the ACM

This paper describes a simple, efficient algorithm to locate all occurrences of any of a finite number of keywords in a string of text. The algorithm consists of constructing a finite state pattern matching machine from the keywords and then using the pattern matching machine to process the text string in a single pass. Construction of the pattern matching machine takes time proportional to the sum of the lengths of the keywords. The number of state transitions made by the pattern matching machine in processing the text string is independent of the number of keywords. The algorithm has been used to improve the speed of a library bibliographic search program by a factor of 5 to 10.

A meta-learning approach to automatic kernel selection for support vector machines

Appropriate choice of a kernel is the most important ingredient of the kernel-based learning methods such as support vector machine (SVM). Automatic kernel selection is a key issue given the number of kernels available, and the current trial-and-error nature of selecting the best kernel for a given problem. This paper introduces a new method for automatic kernel selection, with empirical results based on classification. The empirical study has been conducted among five kernels with 112 different classification problems, using the popular kernel based statistical learning algorithm SVM. We evaluate the kernels' performance in terms of accuracy measures. We then focus on answering the question: which kernel is best suited to which type of classification problem? Our meta-learning methodology involves measuring the problem characteristics using classical, distance and distribution-based statistical information. We then combine these measures with the empirical results to present a rule-based method to select the most appropriate kernel for a classification problem. The rules are generated by the decision tree algorithm C5.0 and are evaluated with 10-fold cross validation. All generated rules offer high accuracy ratings.

Automatic model selection for the optimization of support vector machine kernels

This approach aims to optimize the kernel parameters and to efficiently reduce the number of support vectors, so that the generalization error can be reduced drastically. The proposed methodology suggests the use of a new model selection criterion based on the estimation of the probability of error of the SVM classifier. For comparison, we considered two more model selection criteria: GACV ('Generalized Approximate Cross-Validation') and VC ('Vapnik-Chernovenkis') dimension. These criteria are algebraic estimates of upper bounds of the expected error. For the former, we also propose a new minimization scheme. The experiments conducted on a bi-class problem show that we can adequately choose the SVM hyper-parameters using the empirical error criterion. Moreover, it turns out that the criterion produces a less complex model with fewer support vectors. For multi-class data, the optimization strategy is adapted to the one-against-one data partitioning. The approach is then evaluated on images of handwritten digits from the USPS database.

A particle swarm optimization and pattern search based memetic algorithm for svms parameters optimization

Addressing the issue of SVMs parameters optimization, this study proposes an efficient memetic algorithm based on Particle Swarm Optimization algorithm (PSO) and Pattern Search (PS). In the proposed memetic algorithm, PSO is responsible for exploration of the search space and the detection of the potential regions with optimum solutions, while pattern search (PS) is used to produce an effective exploitation on the potential regions obtained by PSO. Moreover, a novel probabilistic selection strategy is proposed to select the appropriate individuals among the current population to undergo local refinement, keeping a well balance between exploration and exploitation. Experimental results confirm that the local refinement with PS and our proposed selection strategy are effective, and finally demonstrate effectiveness and robustness of the proposed PSO-PS based MA for SVMs parameters optimization.

A hyper-heuristic evolutionary algorithm for automatically designing decision-tree algorithms

Hyper-heuristic evolutionary algorithms (HHEA) are successful methods for selecting and building new heuristics or algorithms to solve optimization or machine learning problems. They were conceived to help answer questions such as given a new classification dataset, which of the solutions already proposed in the literature is the most appropriate to solve this new problem? In this direction, we propose a HHEA to automatically build Bayesian Network Classifier (BNC) tailored to a specific dataset. BNCs are powerful classification models that can deal with missing data, uncertainty and generate interpretable models. The method receives an input a set of components already present in current BNC algorithms and a specific dataset. The HHEA then searches for the best combination of components according to the input dataset. Results show the customized algorithms generated obtain results of F-measure equivalent or better than other state of the art BNC algorithms.

III. PROPOSED WORK

The flowchart of the proposed methodology (abbreviated as HH-SVM) is depicted in Figure 1. The methodology has two parts: the SVM and the hyper-heuristic framework. The main role of the hyper-heuristic framework is to generate a configuration (C, kernel type and kernel parameters) and send it to the SVM. The SVM uses the generated configuration to solve a given problem instance and then sends the cost function (mean values of errand NSV) to the hyper heuristic framework. This process is repeated for a certain number of iterations. In the following subsections, we discuss the proposed hyper-heuristic framework along with its main components.

THE PROPOSED HYPER-HEURISTIC FRAMEWORK

The proposed hyper-heuristic framework for configuration selection is shown in Figure 2. It has two levels: the high level strategy and the low-level heuristics [11]. The high-level strategy operates on the heuristic space instead of the solution space. In each iteration, the high-level strategy selects a heuristic from the existing pool of low-level heuristics, applies it to the current solution to produce a new solution and then decides whether to accept the new solution. The low level heuristics constitute a set of problem-specific heuristics that operate directly on the solution space of a given problem [39]. To address the bi-objective optimisation problem, we propose a population-based hyper-heuristic framework that operates on a population of solutions and uses an archive to save the non-dominated solutions. The proposed framework combines the strengths of decomposition- and Pareto (dominance)- based approaches to effectively approximate the Pareto set of SVM configurations. Our idea is to combine the diversity ability of the decomposition approach with the convergence power of the dominance approach. The decomposition approach operates on the population of solutions, whereas the dominance approach uses the archive. The hyper heuristic framework generates a new population of solutions using either the old population, the archive, or both the old population and the archive. This allows the search to achieve a proper balance between convergence and diversity. It should be noted that seeking good convergence involves minimising the distances between the solutions and PF, whereas seeking high diversity involves maximising the distribution of the solutions along PF.

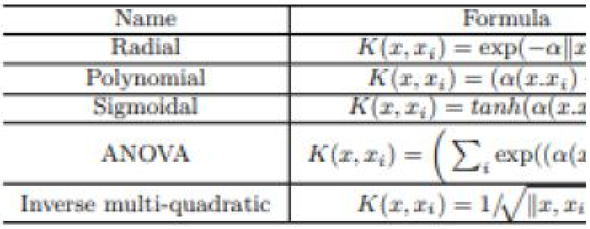
ALGORITHM

Support vector machines:

SVMs are a class of supervised learning models that have been widely used for classification and regression. SVMs are based on statistical learning theory and are better able to avoid local optima than other classification algorithms. An SVM is a kernel-based learning algorithms that seeks the optimal hyperplane. The kernel learning process maps the input patterns into a higher-dimensional feature space in which linear separation is feasible.The basic idea of the SVM approach is to map the input vector xi into an Ndimensional feature space and then construct the optimal decision-making function in the feature space as follows weight vector; C is the margin parameter (or penalty); The existing kernel functions can be classified as either local or global kernel functions [9]. Local kernel functions have a good learning ability but do not have a good generalisation ability. By contrast, global kernel functions have a good generalization ability but a poor learning ability. For example, the radial kernel function is known to be a local function, whereas the polynomial kernel function is a global kernel function. The main challenge lies in determining which kernel function should be used for the current problem instance or the current decision point. This is because the kernel selection process strongly depends on the distribution of the input vectors and the relationship between the input vector and the output vector (predicted variables). However, the feature space distribution is not known in advance and may change during the course of the solution process, especially in big data cyber security. Consequently, different kernel functions may work well for different instances or in different stages of the solution process, and kernel selection may thus have a crucial impact on SVM performance. To address this issue, in this work, we use multiple kernel functions to improve the accuracy of our algorithm and avoid the shortcomings of using a single kernel function.

Table.1 Kernal Functions

ISSN: 2366-1313



Name	Formula
Radial	$K(x, x_i) = \exp(-\alpha \| x$
Polynomial	$K(x, x_i) = (\alpha(x.x_i)$
Sigmoidal	$K(x, x_i) = tanh(\alpha(x.x$
ANOVA	$K(x, x_i) = \left(\sum_i \exp((\alpha(x\right.$
Inverse multi-quadratic	$K(x, x_i) = 1/\sqrt{\| x, x_i}$

The project is carried out based on the following modules listed:

Approved Users

In this system users are not allowed to access resources simply. User need verify their information's with admin. Admin are the authorized and trustworthy to the network. User need to send the request to administrator that they are interested to add the community. Admin views the user request and respond with the pass code to access users account through trusted sources like SSL (Gmail).

Security Steps and Upload

This is where the proposed algorithm is going to be effective. The admin can be uploading the files with proposed classification algorithm and cryptography in order to classify and upload the encrypted details to network with its tag in the mark of understand to user about the resource.

Resource Access

The permissions to access the resource can be sent by users to admin. The requests have been updated by admin with the response to access the resource. Users can decrypt the resource and access the details. The important part is accessing the resource with the decryption. The passkey to access the details are limited. If the limit of wrong attempts over the threshold value means pass key expires.

Graphical Representation

This is graphical notation of the data given by the system. This phase of implementation will shows the effectiveness of the proposed system through pictorially in the order to better understand of proposed system.

SYSTEM DESIGN

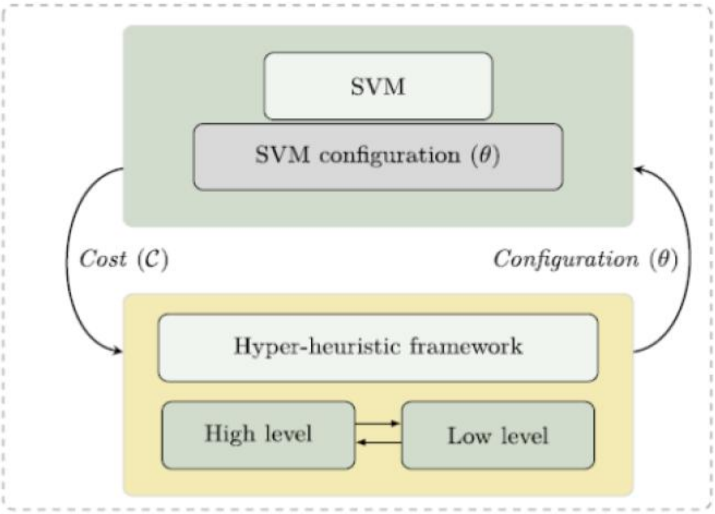


Fig.1 System architecture

The system architecture has two parts: the SVM and the hyper-heuristic framework. The main role of the hyper-heuristic framework is to generate a configuration (C, kernel type and kernel parameters) and send it to the SVM. The SVM uses the generated configuration to solve a given problem instance and then sends the cost function (mean values of err and NSV) to the hyper-heuristic framework. This process is repeated for a certain number of iterations. In the following subsections, we discuss the proposed hyper heuristic framework along with its main component.

IV. RESULTS

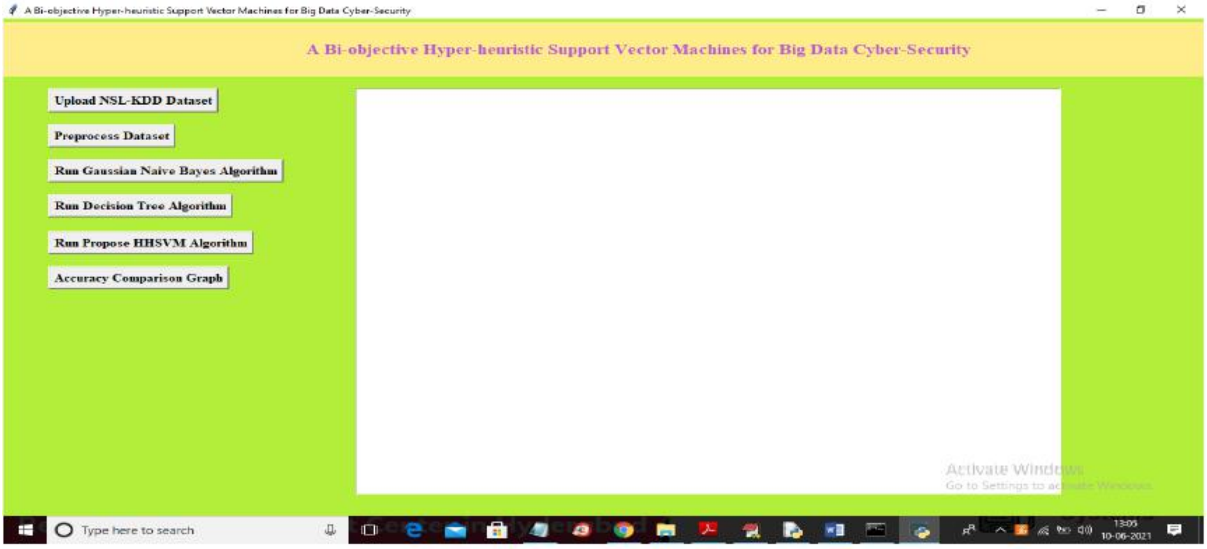
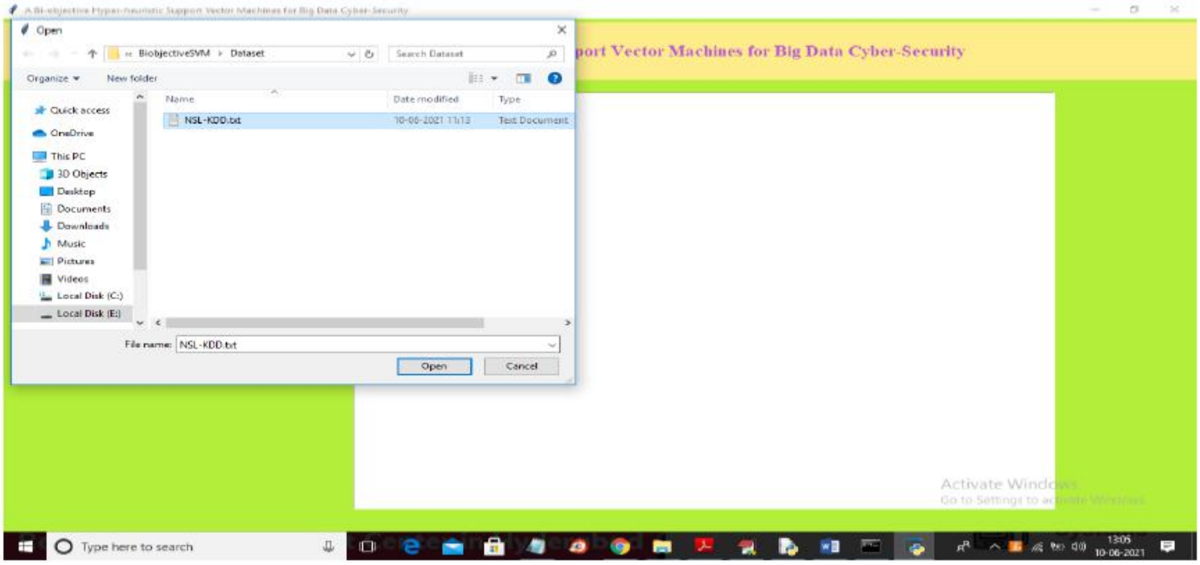

A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
Upload NSL-KDD Dataset
Preprocess Dataset
Run Gaussian Naive Bayes Algorithm
Run Decision Tree Algorithm
Run Propose HHSVM Algorithm
Accuracy Comparison Graph
Activate Windows
Go to Settings to activate Windows.
Type here to search
13:05
10-06-2021

Fig.2 uploading NSL- KDD Dataset

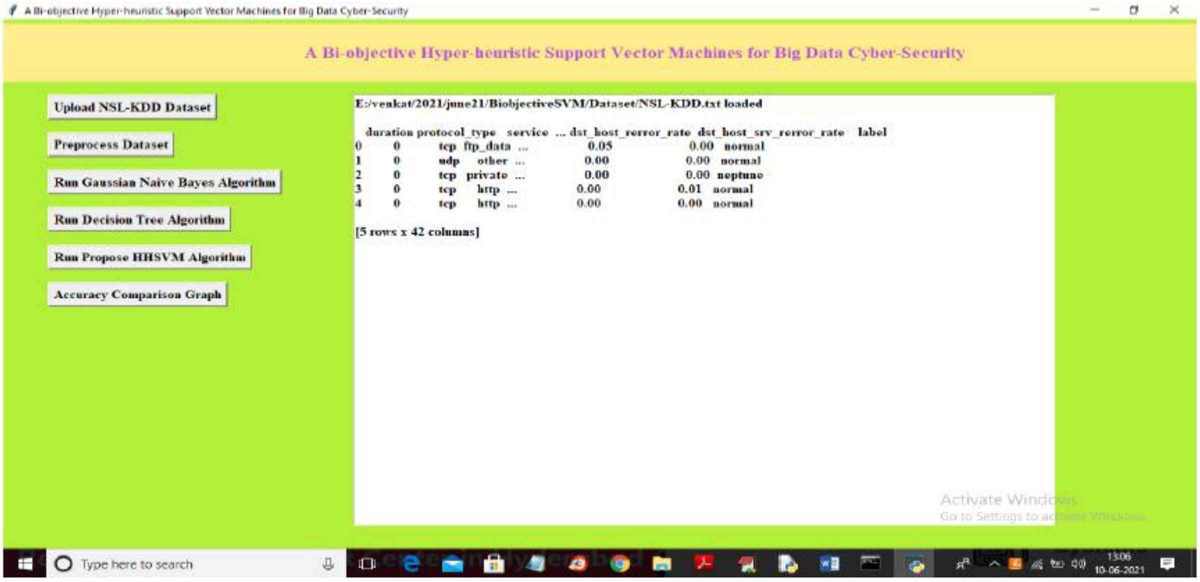
In above screen click on 'Upload NSL-KDD Dataset' button to upload dataset and to get below screen



Name	Date modified	Type
NSL-KDD.txt	10-06-2021 11:13	Text Document

File name: NSL-KDD.txt

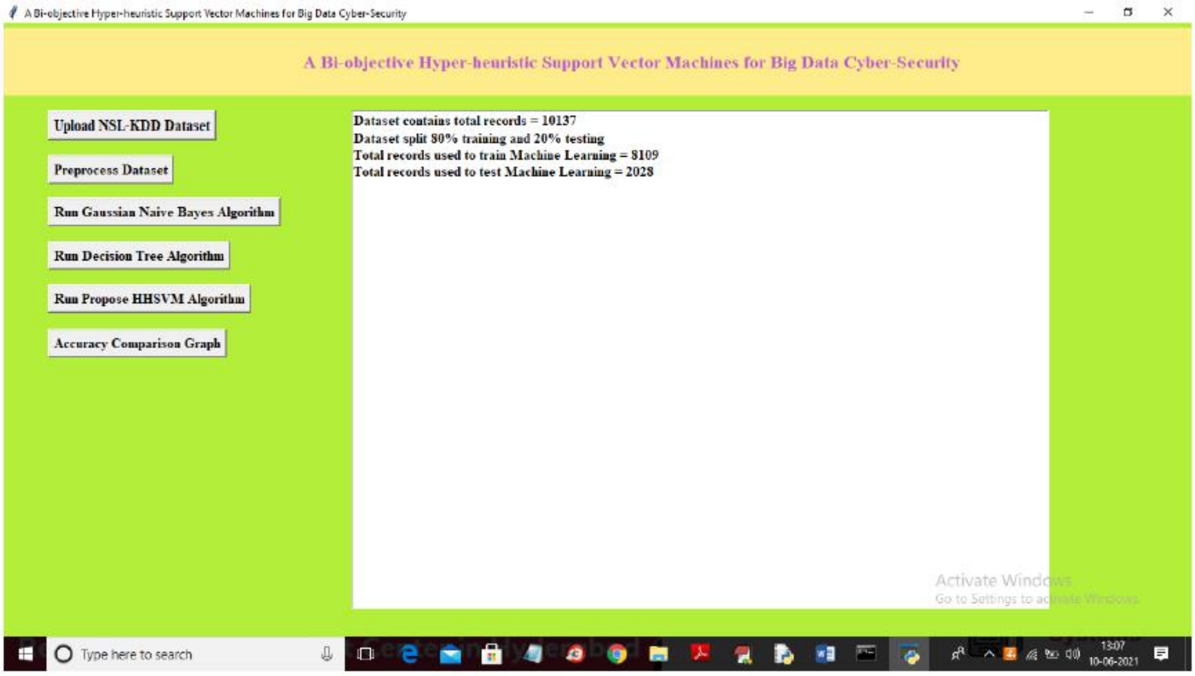
Fig.3 In above screen selecting and uploading ‘NSL-KDD.txt’ dataset and then click on ‘Open’ button to load dataset and to get below screen



A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
Upload NSL-KDD Dataset
Preprocess Dataset
Run Gaussian Naive Bayes Algorithm
Run Decision Tree Algorithm
Run Propose HHSVM Algorithm
Accuracy Comparison Graph
Activate Windows
Go to Settings to activate Windows.
Type here to search
13:06
10-06-2021

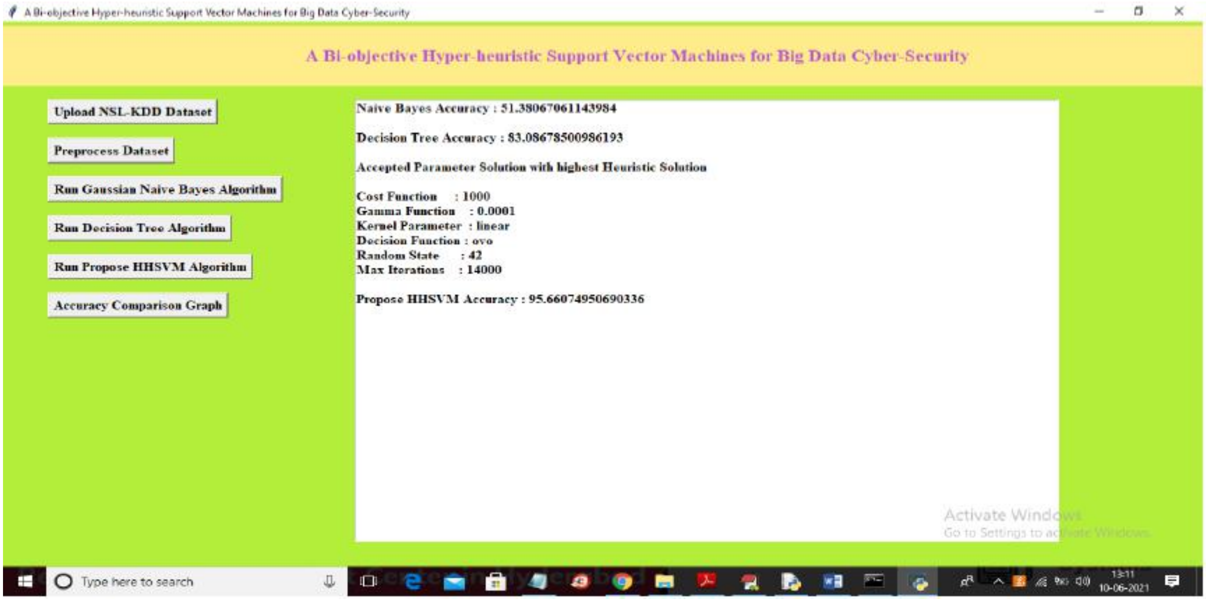
```
E:/venkat/2021/june21/BiobjectiveSVM/Dataset/NSL-KDD.txt loaded

   duration protocol_type   service  ... dst_host_rerror_rate  dst_host_srv_rerror_rate   label
0         0           tcp  ftp_data  ...                 0.05                      0.00  normal
1         0           udp     other  ...                 0.00                      0.00  normal
2         0           tcp   private  ...                 0.00                      0.00  neptune
3         0           tcp      http  ...                 0.00                      0.01  normal
4         0           tcp      http  ...                 0.00                      0.00  normal

[5 rows x 42 columns]
```

Fig.4 In above screen dataset loaded and I am displaying few records from dataset and in above dataset we can see some values are non-numeric and machine learning will not accept non-numeric values so we need to pre-process those values to assign integer id to each unique non-numeric value


A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
Upload NSL-KDD Dataset
Preprocess Dataset
Run Gaussian Naive Bayes Algorithm
Run Decision Tree Algorithm
Run Propose HHSVM Algorithm
Accuracy Comparison Graph
Type here to search
13:07
10-06-2021

```
Dataset contains total records = 10137
Dataset split 80% training and 20% testing
Total records used to train Machine Learning = 8109
Total records used to test Machine Learning = 2028
```

Fig.5 In above screen dataset is pre-processed and dataset contains huge 10137 records where application using 8109 records for training and 2028 records testing trained ML model accuracy. After train model test records will apply on trained model to perform prediction and then correct prediction percentage will be consider as accuracy.


A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
Upload NSL-KDD Dataset
Preprocess Dataset
Run Gaussian Naive Bayes Algorithm
Run Decision Tree Algorithm
Run Propose HHSVM Algorithm
Accuracy Comparison Graph
Type here to search
13:11
10-06-2021

```
Naive Bayes Accuracy : 51.38067061143984

Decision Tree Accuracy : 83.08678500986193

Accepted Parameter Solution with highest Heuristic Solution

Cost Function     : 1000
Gamma Function    : 0.0001
Kernel Parameter  : linear
Decision Function : ovo
Random State      : 42
Max Iterations    : 14000

Propose HHSVM Accuracy : 95.66074950690336
```

Fig.6 In above screen Naïve Bayes got 51% accuracy, decision tree we got 83% accuracy, optimize parameters we got 95% accuracy for HHSVM algorithm, we can see by applying

‘Bi-objective Hyper-heuristic’ technique for SVM we got high accuracy and now click on ‘Accuracy Comparison Graph’ button to get below graph

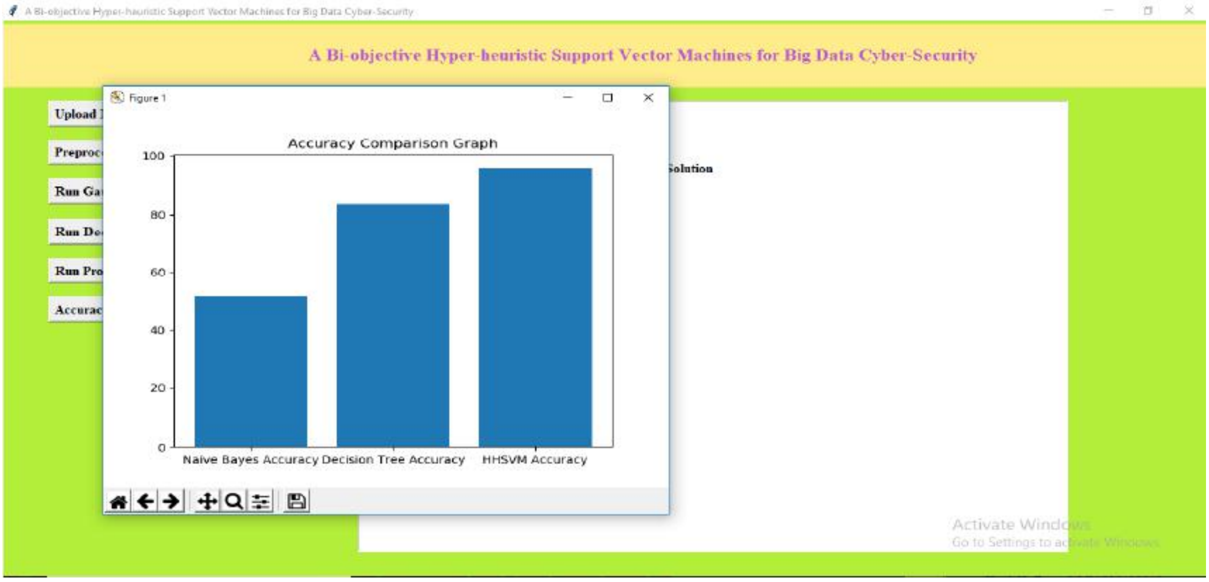

A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
A Bi-objective Hyper-heuristic Support Vector Machines for Big Data Cyber-Security
Upload I
Preproc
Run Ga
Run De
Run Pro
Accurac
Figure 1
Accuracy Comparison Graph
100
80
60
40
20
0
Naive Bayes Accuracy
Decision Tree Accuracy
HHSVM Accuracy
Solution
Activate Windows
Go to Settings to activate Windows.

Fig.7 In above graph x-axis represents algorithm name and y-axis represents accuracy of those algorithms and from above graph we can conclude that HHSVM got high accuracy

V. CONCLUSION

In this work, we proposed a hyper-heuristic SVM optimisation framework for big data cyber security problems. We formulated the SVM configuration process as a bi-objective optimisation problem in which accuracy and model complexity are treated as two conflicting objectives. This bi-objective optimisation problem can be solved using the proposed hyper-heuristic framework. The framework integrates the strengths of decomposition- and Pareto-based approaches to approximate the Pareto set of configurations. Our framework has been tested on two bench-mark cyber security problem instances: Microsoft malware big data classification and anomaly intrusion detection. The experimental results demonstrate the effectiveness and potential of the proposed framework in achieving competitive, if not superior, results compared with other algorithms .

REFERENCES

[1] Mansour Ahmadi, Dmitry Ulyanov, Stanislav Semenov, Mikhail Trofimov, and Giorgio Giacinto. Novel feature extraction, selection and fusion for effective malware family classification. In Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, pages 183–194. ACM, 2016.

[2] Alfred V Aho and Margaret J Corasick. Efficient string match- ing: an aid to

bibliographic search. Communications of the ACM, 18(6):333–340, 1975.

[3] Shawkat Ali and Kate A Smith-Miles. A meta-learning ap- proach to automatic kernel selection for support vector machines. Neurocomputing, 70(1):173–186, 2006.

[4] Nedjem-Eddine Ayat, Mohamed Cheriet, and Ching Y Suen. Automatic model selection for the optimization of support vec- tor machine kernels. Pattern Recognition, 38(10):1733–1745, 2005.

[5] Yukun Bao, Zhongyi Hu, and Tao Xiong. A particle swarm optimization and pattern search based memetic algorithm for svms parameters optimization. Neurocomputing, 117:98–106, 2013.

[6] Rodrigo C Barros, M arcio P Basgalupp, Andr e CPLF de Car- valho, and Alex A Freitas. A hyper-heuristic evolutionary algorithm for automatically designing decision-tree algorithms. In Proceedings of the 14th annual conference on Genetic and evolutionary computation, pages 1237–1244. ACM, 2012.

[7] M arcio P Basgalupp, Rodrigo C Barros, Tiago S da Silva, and Andr e CPLF de Carvalho. Software effort prediction: a hyper- heuristic decision-tree based approach. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, pages 1109–1116. ACM, 2013.

.[8] M arcio P Basgalupp, Rodrigo C Barros, and Vili Podgor- elec. Evolving decision-tree induction algorithms with a multi- objective hyper-heuristic. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, pages 110–117. ACM, 2015.

[9] Asa Ben-Hur and Jason Weston. A users guide to support vector machines. Data mining techniques for the life sciences, pages 223–239, 2010.