Multi-Objective Hyper-Heuristic Improved Particle Swarm Optimization Based Configuration of Support Vector Machines for Big Data Cyber Security

Aswanandini.R, Muthumani.N

Abstract- The massive increase of information in the big data era has not only created data processing problems, but also the data security issues. These big data cyber security issues can be handled effectively using machine learning algorithms among which the Support Vector Machines (SVM) has better results on big data classification problems. Defining the proper configuration of the SVM requires expert knowledge in selecting the kernel function and other parameters and this can significantly improve its classification results. In this paper, the SVM configuration process is modelled as a multi-objective optimization problem by considering the false positive rate, false negative rate and model complexity parameters. A Hyper-Heuristic Improved Particle Swarm Optimization (HHIPSO) framework is developed to optimize the SVM multi-objective optimization problem by incorporating the hyper-heuristics and improved particle swarm optimization algorithm. The proposed hyper-heuristic framework includes the high-level strategy for controlling the selection of low-level heuristics by search process and the low-level heuristics generate the new SVM configuration solutions using different rules of PSO. The effective selection of the kernel function and the respective parameters of the SVM should result in better values of false positive rate and false negative rate and also reduce the complexity. The evaluation of the proposed HHIPSO is performed on two cyber security problems and the obtained results illustrated that the proposed approach is effective in improving the classification of big data cyber security problems than the other algorithms.

Keywords: Big data, cyber security, Support Vector Machines, multi-objective optimization, hyper-heuristics, Hyper-Heuristic Improved Particle Swarm Optimization.

I. INTRODUCTION

Modern digital information era has created the space for high volume of data to be generated and stored by the advanced technologies and Internet of Things (IoT) [1]. This rapid growth of the Internet data has also exponentially increased the frequency of cyber-attacks. The cyber-attacks cause extensive damages to the networks and hence to tackle them the cyber security systems have been designed and installed. Cyber security techniques and processes are assigned with the role of thwarting the illegal cyber-attacks to protect the computers and networks from the cyber damages [2]. They perform the major function of protecting the shared information for improving decision making; detecting the vulnerable attacks in applications; prevent unauthorized accessing of networks and secure the confidential network information [3]. Most of the larger companies have their own cyber security network while other organizations make use of such solutions from security organizations like Accenture, IBM, CISCO, etc. [4].

Recent cyber security solutions have inclined more towards the monitoring of network and Internet traffic to identify and avert the bad actions [5]. This is entirely different from the traditional cyber security solutions which focus only on the detection of bad signatures for unauthorized access. While the traditional systems were aimed at detecting the malware by scanning the incoming traffic against the malware signatures, they are relatively weaker with detecting only limited threats [6]. These traditional techniques including the intrusion detection, firewalls and anti-virus software have become ineffective in tackling the hackers as the attack strategies are highly destructive than the older versions [7]. In addition to this, the presence of big data has increased the critical condition as gigabytes of data are transferred between each node of the computer networks; making the hackers job of entering the networks very easier and cause severe damage without getting traced [8]. The big data problems are majorly due to the organizations providing access to their data networks allowing the partners and consumers to access all data and making it vulnerable to the cyber-attacks. Similarly, the big data has also increased the skills of hackers to evade the traditional security systems. Also, the big data has made it difficult to identify the attacks when initiated and the attack is only known after the damage is done to the hardware and software components [9].

To address these security threats linked to the big data, the big data analytics can be used for cyber security analytics by employing the big data techniques to evade the cyber-attacks [10].

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author

Aswanandini.R*, M.Sc.,M.Phil., Ph.D Scholar, Sri Ramakrishna College of Arts and Science,Assistant Professor, KG College of Arts and Science, Coimbatore.aswanandini1981@gmail.com

Dr.Muthumani.N. Ph.D, Professor & Head, Department of Mathematics(CA), Sri Ramakrishna College of Arts and Science, Coimbatore. muthumani@srcas.ac.in,

Based on this concept, many organizations have started to remodel the cyber security systems [11]. As said before, the machine learning algorithms have been utilized extensively for this process with the SVM emerging as the front-runner.

Recently, in [12], SVM configuration process was modelled as a bi-objective optimization problem and a hyper-heuristic framework was developed to optimize this cyber security problem. However, the modelling of SVM configuration as bi-objective problem considers only accuracy and model complexity factors. Considering multiple parameters can be beneficial in big data classification problems. Likewise, the hyper-heuristic framework can also be improved further if advanced optimization concepts are adapted to its basic structure. Hence, in this paper, the SVM configuration is modelled as multi-objective optimization problem and an efficient HHIPSO is proposed to resolve the optimization problem. First, the false positive rate, false negative rate and model complexity parameters are considered for the modelling of multi-objective optimization problem. Then the hyper-heuristic framework controls the selection of kernel function and related parameters of SVM. Then the HHIPSO adaptively analyses and detects the suitable set of SVM configurations. The performance of the proposed HHIPSO based framework is evaluated using two cyber security problems: NSL-KDD anomaly intrusion detection and ISCX-IDS Intrusion detection. The observed results illustrate the effectiveness of the proposed hyper-heuristic framework in the cyber security problems.

The rest of the article is organized as: Section 2 presents a brief discussion of the recent related works. Section 3 presents the formulation of the SVM configuration and section 4 presents the proposed hyper-heuristic framework. The experimental results are highlighted in section 5 while the conclusion of this paper is provided in section 6.

II. RELATED WORKS

Many researchers have focused on developing efficient cyber security solutions using big data analytics. Some of the most recent techniques are discussed in this section. Dovom et al. [13] presented a fuzzy pattern tree method for malware detection of big data IoT. This method transmutes the Op-codes into vector space and applies y fuzzy and fast fuzzy pattern tree to detect the malwares. The results provided high degree of accuracy in categorization with about 97% for Kaggle dataset and more than 93.13% for Ransomware dataset. Shamshirband and Chronopoulos [14] developed high performance-ELM based malware detection method which provided accuracy of 95.92%. However, this model does consider only three features for malware detection and hence needs to be improved. Zhong and Gu [15] proposed a multi-level deep learning system detecting the malwares. This system organizes multiple deep learning models using the tree structure and each tree focuses on specific data distribution of particular malware group. Experimental results showed high accuracy of malware detection using this system but the major drawback is the high computation time. Ju et al. [16] proposed a big data analytics framework for targeted cyber-attacks detection from the heterogeneous noisy data. This approach utilized different heterogeneous data and correlated them to identify the malicious nodes. It provides highly accurate cyber-attack detection but it considers only limited features and does not include the human perception in attack detection.

Venkatraman et al. [17] introduced a hybrid deep learning image-based analysis model for detecting the cyber-attacks. This hybrid model helps in detecting suspicious behavior of systems and also visualizes the malware classification. This model achieves high accuracy of malware detection with less computational cost. Calvert and Khoshgoftaar [18] used the big data sampling to produce varying class distributions for the detection of slow HTTP DoS attacks. This approach is based on the legitimate traffic monitoring of the system to detect the attacks using Random forest as optimal learning algorithm. This approach provided results of AUC value 0.99904 for the attack detection. However, only the AUC metric is estimated and this causes statistical inconsequential decisions. Mao et al. [19] presented a spatio-temporal approach to detect the malwares based on the big data characteristics of the cloud systems. This approach devised a graph based semi-supervised learning algorithm for detecting the attacks based on the spatial and temporal features of the data distributions. Experimental results provided better detection rate of malwares in less computation time. But in this approach there is an upper bound on the recall to malware detection based on the file co-occurrence in end hosts.

Martín et al. [20] introduced MOCDroid using multi-objective evolutionary classifier detecting the malwares in Android. This approach utilizes SPEA2, a multi-objective genetic algorithm, to select groups of import terms to determine the malware nodes. Empirical results proved that this approach has high accuracy and reduced number of false positives; but the approach considers only few objectives. Gupta and Rani [21] proposed machine learning based big data framework for zero-day malware detection. The identification of attacks is performed using classification algorithms in which the random forests provided higher accuracy. However, the larger dataset used for evaluation makes the detection process very slow. Wassermann and Casas [22] developed BIGMOMAL method using big data analytics and supervised-machine-learning for mobile malware detection. This approach detected the malware in running apps with high accuracy but the approach suffers from concept drift problem. From the literature, it can be understood that the machine learning algorithms can provide better classification in the detection of malwares. However, it is also inferred that certain classifiers are only suitable for particular type of datasets. This leads to the necessity of the designing better configuration of the machine learning algorithms which could provide highly accurate malware detection with less computation time and higher efficiency.

III. SVM CONFIGURATION AND OBJECTIVE FUNCTION FORMULATION

SVMs are supervised kernel learning based algorithms whose primary role was classification and regression. The kernel learning maps the input patterns into the higher dimensional feature space for linear separation. Hence the kernel function and the kernel parameters must be selected appropriately to improve the performance of SVM. The radial, polynomial, sigmoidal, ANOVA and inverse multi-quadratic are some of the kernel functions used for SVM. Many researchers have developed new and hybrid kernel functions by combining the basic kernels. The existing kernels are either local or global kernels. The local kernels have good learning ability but have poor generalization ability while global kernels have good generalization but poor learning ability. The main challenge is to select the kernel function which should be used for the current problem instance or the current decision point. The selection is based on the distribution of the input vectors and the relationship between the input vector and the output vector. However, the feature space varies during the solution process and hence it is determined to use multiple kernels from which the best suitable kernel is selected for each instance. This approach improves the accuracy of SVM but the selection of kernel must be automatic and appropriate to the current stage of processing.

The kernel functions are formulated by the input vectors and kernel parameters α, β and d which are set by the user. Apart from these parameters, the weight vector w and the margin parameter C area also set by the user. In traditional SVM configuration, the appropriate values for *C*, the kernel type and the kernel parameters are needed to be specified. The objective is to select SVM configuration that minimizes the error and improves the accuracy without influencing the complexity in any tested data. This can be modelled as a black-box optimization problem expressed as a tuple form

$$< SVM, \Theta, \mathcal{D}, \mathcal{C}, \mathcal{S} > \qquad (1)$$

Where SVM is the configured algorithm, Θ is the search space of the possible SVM configurations (*C*, kernel type and kernel parameters), $\mathcal{D}$ is the distribution of the set of instances, $\mathcal{C}$ is the cost function, and $\mathcal{S}$ is the statistical information. The goal is to minimize the cost function $\mathcal{C}$ to obtain the solution sets over a set of problem instances to find

$$\theta^* \in \underset{\theta \in \Theta}{\mathrm{ARG\,MIN}} \frac{1}{|\mathcal{D}|} \cdot \sum_{\pi \in \mathcal{D}} \mathcal{C}(\theta, \pi) \qquad (2)$$

Each $\theta \in \Theta$ represents one possible configuration of the SVM and the output $\mathcal{C}$ is obtained while testing SVM across many instances. The main function of the proposed HHIPSO will be to find $\theta \in \Theta$ such that the cost function is optimized.

The multi-objective optimization problem is formulated based on the pre-determined three objective parameters. For an SVM, the accuracy can be seen as a trade-off between the complexity i.e. number of support vectors (*NSV*) and the margin (*C*). This trade-off is controlled through the selection of the SVM configuration (*C*, kernel type and kernel parameters). To ensure this, the false positive rate (fpr), false negative rate (fnr) and model complexity are selected as the conflicting objectives. fpr and fnr are the expectancy rates related to the accuracy, precision, recall and f-measure values. The complexity is represented by the number of support vectors (*NSV*). The multi-objective optimization problem is formulated as

$$\min F(X) = |f_1(x), f_2(x), f_3(x)| \qquad (3)$$

Where $f_1(x) = fpr; f_2(x) = fnr; f_3(x) = NSV$

Resolving this formula provides the solution for SVM configuration and this will be the main function of the proposed multi-objective optimization framework.

IV. HHIPSO FRAMEWORK

The proposed HHIPSO framework consists of the SVM and the hyper-heuristic framework containing the IPSO algorithm for multi-objective optimization. Fig.1 shows the proposed HHIPSO based SVM working model. The proposed model performs the processes of cost optimization based configuration selection for the SVM. The SVM model includes the SVM configuration and setting of kernel function and kernel parameters along with the margin and other parameters. The cost function modelled using the SVM configuration parameters is to be optimized using the multi-objective optimization function. This function is based on the fpr, fnr and model complexity and is performed using the hyper-heuristics and the IPSO.

The solution representation is the one configuration ($\theta \in \Theta$) of the SVM that is optimal and is represented as a one-dimensional array with the margin, kernel type and kernel parameters to be selected. The population of the IPSO is randomly initialized by assigning a random value to each decision variable.

$$x_i^p = l_i^p + Rand_i^p(0,1) \times \left(u_i^p - l_i^p\right), \quad p = 1,2,\dots,|PS|; i = 1,2,\dots,d \qquad (4)$$

Where i denotes the index of the decision variable, *d* represents the total number of decision variables, *p* denotes the index of the solution, |*PS*| denotes the population size, $Rand_i^p(0,1)$ denotes the random value in the range [0,1] for the *i-th* decision variable, l_i^p and u_i^p are the linear lower and upper bounds, respectively.

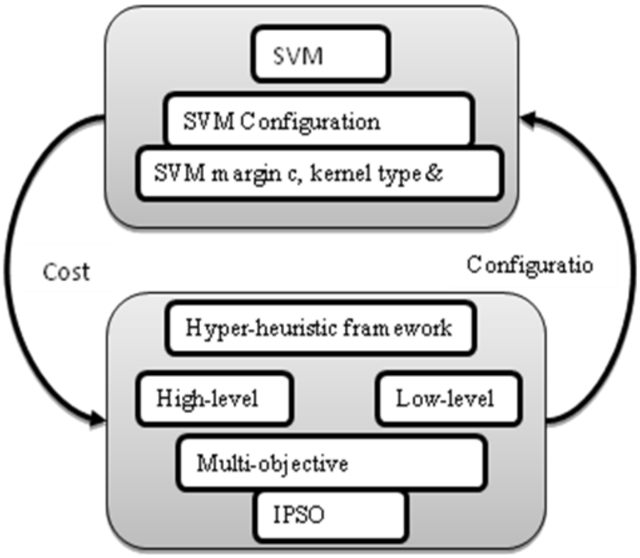

Fig.1. Proposed HHIPSO-SVM model

The fitness calculation is performed in the IPSO by estimating the generated rules to optimize the cost function based on the objective function in Eq. (3).

This will be applied to all the training instances T and determine the effective configuration with minimal cost function. The fitness function is first decomposed into a number of single-objective sub-problems which are solved in a collaborative manner to form the objective values. The fitness function is modelled as

$$fitness(X) = |\frac{1}{T}| \sum_{i=1}^{|T|} z_i(X) \qquad (5)$$

Subject to $z_i(X) = \min F(X)$

Where $z_i(X)$ is the objective value obtained for the instance i in the training set T using the set of generated rules.

High-level strategy: The primary function of the high-level strategy is to automatically perform the heuristic selection by choosing the heuristics one-by-one and applying it to the solutions. In the selection stage, the heuristics are chosen from the existing set of heuristics formed by the low-level heuristics. They are chosen using a online heuristic selection mechanism by analysing its past performance for using it in the current state. In this selection, two variables are vital. They are the empirical reward and the confidence level. The rewards obtained in past performance are called empirical reward while the frequency of utilization of the heuristic denotes the confidence level. Based on these two variables, the heuristics is deemed fit or unfit for the current state of operation.

Once the heuristics are selected, they are applied on the solutions. First the solution to which the heuristic is to be applied in the particle mating process is selected. The solution is selected based on the particle position and velocity of the IPSO algorithm. The particle which is provided as $gbest$ solution contains the solution to be applied. The heuristic is applied with the selected solution to form new set of solutions. The new solutions are compared and they are analysed by their properties in terms of configuration to either include them in the existing set of solutions or terminate them to accommodate newer solution from next iterations.

Low-level heuristics: The low-level heuristics contains the set of problem related rules generated to provide solutions to each selected problem instances. The low-level heuristics considers one or more solutions and either combines or modifies them to form new set of solutions. The solutions are formed using many search based operations. The IPSO based search process is one of the search processes utilized to form new solutions from the existing set of solutions.

After the formation of new solutions by low-level heuristics and the selection by the high-level strategy, they are saved in the non-dominated set of solutions in the archive. The non-dominated sorting procedure is used to classify the archive to create several levels for saving the newer solutions. The first level is given to the solution with high priority and the next level will be given to the second best priority and vice versa. The IPSO selects the solutions from this archive based on the Pareto-front and returns the best configuration as the final solution.

IPSO: The proposed enhancement to the PSO algorithm is to improve the local search capacity. The velocity and position update of PSO is given by

$$v_{id} = w * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \qquad (6)$$

$$x_{id} = x_{id} + v_{id} \qquad (7)$$

Here the v_{id} denotes the velocity of particle i at D-dimensional vector, x_{id} denotes the position of particle i at D-dimensional vector, w is the inertia weight, c_1 and c_2 are positive acceleration factors, and r_1 and r_2 are random functions varying between [0, 1]. p_{id} and p_{gd} are local optimal extreme and global optimal extreme, respectively. In traditional PSO algorithm, the global search ability is decreased with the increase in the number of iterations. In order to tackle this problem, the local minimum value is slimmed down to enhance the global search ability. This can create the solution to be struck in the local minimum value and also reduces the convergence speed.

The proposed IPSO aims to overcome this limitation by increasing the inertia weight at each iteration and exchange the high frequency non-linear function as the inertia weight attenuation function to improve the convergence speed. Thus the velocity is modified based on the new inertia weight $w(t)$ is formulated as

$$v_{id} = w(t) * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \qquad (8)$$

$$w(t) = w_{max} - \left(\frac{w_{max} - w_{min}}{2^{\frac{t}{T_{max}}}} \right) * \left(\frac{t}{T_{max}} \right)^2 \qquad (9)$$

Here $w(t)$ denotes the inertia weight after t iteration, T_{max} is the maximum iteration, and w_{max} and w_{min} are the maximum and minimum inertia weights. When the initial t value is small, the inertia weight is close to w_{max} and increases the global search ability. With the increase in the number of iterations, the inertia weight stays balanced and increases the local search ability and avoids the local minimum value, thus improving the convergence speed. Algorithm 1 shows the procedure of the proposed HHIPSO framework.

Algorithm 1: HHIPSO

```
i. Begin
ii. Select set of training instances T
iii. Generate rules for the selected problem instance
iv. Initialize the population randomly
        $X = \{x_1, x_2, \dots . x_N\}$
v. Iteration $t = 0$
vi. While $t \leq T_{max}$ do
        a. For each $x_i \in X$ do
                i. Compute $fitness(X)$ using Eq. (5)
                ii. Solve problem instance $T_k \in T$
                iii. Estimate values of C, kernel & parameters
        b. End for
```

```
vii. Update $pbest$ and $gbest$
viii. Update velocity and position of each particle by Eq. (7) & (8)
ix. $t = t + 1$
x. End while
xi. Return $gbest$
xii. End
```

V. EXPERIMENTS AND RESULTS

5.1. Experimental setup and benchmark instances

The evaluation of the proposed HHIPSO-SVM model is performed using two benchmark instances of cyber security problems, NSL-KDD anomaly intrusion detection and ISCX-IDS Intrusion detection. The experiments are conducted in MATLAB 2016b (version 9.1) on a Windows 64 bit machine of processor Intel core i5 3470 3.2 GHz, RAM 4GB DDR3 and Storage of 500GB Intel SSD. The two benchmark instances are collected from https://www.unb.ca/cic/datasets/index.html.

A) NSL-KDD Anomaly intrusion detection: The first evaluation is carried out using NSL-KDD data instances. The NSL-KDD consists of training, testing, 20% training and 20% testing data files. It also contains subset file with difficulty levels. The NSL-KDD is an improved version of the popular KDDCUP99 dataset. NSL-KDD problem instance consists of 311,027 training samples and 77,289 testing samples which are classified as either normal or malicious.

B) ISCX-IDS Intrusion detection: ISCX-IDS was created by monitoring the network activity for 7 days from Friday 11/6/2010 to Thursday, 17/6/2010. It consists of records of normal, HTTP Denial of Service attacks, Brute Force attacks and infiltration activities. Around 208,667 training samples and 78,400 testing samples that are classified as either normal or attack activities are used for this evaluation.

5.2. Performance evaluation

The evaluation of the proposed HHIPSO-SVM is done on the two benchmark instances and then the performance is compared with the existing HH-SVM [12]. The comparison metrics are accuracy, precision, recall, f-measure, model complexity (NSVs) and time complexity. Table 1 shows the accuracy comparison of HH-SVM and the proposed HHIPSO-SVM for 25 independent runs. It can be seen that the accuracy values of HHIPSO-SVM are significantly higher than the HH-SVM for both the benchmark instances. For NSL-KDD instance, the accuracy of the proposed framework is higher by around 4% while for the ISCX-IDS instance, it is increased by 5.8%.

Table.1. Accuracy (%) comparison

Algorithm / Instance	NSL-KDD	ISCX-IDS
HH-SVM	89.76	86.6
HHIPSO-SVM	93.33	92.4

The comparison in terms of precision, recall and f-measure are shown in Tables 2, 3 and 4, respectively. The precision results from Table 2 shows that for the NSL-KDD, the HHIPSO-SVM has high precision which is 6.8% greater than HH-SVM while for the ISCX-IDS, HHIPSO also has 6.3% greater precision. Similarly, for recall and F-measure, the values of HHIPSO are significantly greater than the HH-SVM values for both the benchmark instances.

Table.2. Precision (%) comparison

Algorithm / Instance	NSL-KDD	ISCX-IDS
HH-SVM	67.10	63.3
HHIPSO-SVM	73.99	69.65

Table.3. Recall (%) comparison

Algorithm / Instance	NSL-KDD	ISCX-IDS
HH-SVM	62.81	60.0
HHIPSO-SVM	64.29	61.1

Table.4. F-measure (%) comparison

Algorithm / Instance	NSL-KDD	ISCX-IDS
HH-SVM	62.22	56.19
HHIPSO-SVM	68.37	59.82

Table 5 and 6 shows the comparison of HH-SVM and HHIPSO-SVM in terms of model complexity i.e. NSV and time complexity, respectively. For NSL-KDD dataset, HHIPSO-SVM has 11 support vectors compared to the 16 support vectors of HH-SVM. Likewise for ISCX-IDS dataset, HHIPSO-SVM has 23 NSV compared to 34 NSVs of HH-SVM. This justifies the fact that the proposed IPSO based heuristic framework reduces the complexity of the system with its precise and efficient design strategy. Similarly, the time complexity of HHIPSO-SVM is greatly reduced than the HH-SVM model which is evident from the minimal time consumption in Table 6.

Table.5. NSV comparison

Algorithm / Instance	NSL-KDD	ISCX-IDS
HH-SVM	16	34
HHIPSO-SVM	11	23

Table.6. Time complexity (seconds) comparison

Algorithm / Instance	NSL-KDD	ISCX-IDS
HH-SVM	4.65	126
HHIPSO-SVM	2.55	49.5

The performance of the proposed HHIPSO-SVM is also compared with other popular algorithms. The algorithms namely Gaussian Naive Bayes Tree (GNBT) [23], Fuzzy Classifier (FC) [24] and Decision Tree (DT) [25] are used for comparison. The comparison is made in terms of accuracy for the NSL-KDD dataset and it is shown in Table 7.

Table.7. Comparison of accuracy results of HHIPSO-SVM and other algorithms

Algorithm	**NSL-KDD**
DT	80.14

FC	82.74
GNBT	82.02
HH-SVM	89.76
HHIPSO-SVM	**93.33**

From Table 7, it can be inferred that the proposed HHIPSO-SVM framework has better performance than the other algorithms. The main reason for this improvement is the use of better design strategy and efficient solution for different problem instances by selecting the most appropriate SVM configuration at each stage of the system without degrading the data processing performance.

VI. CONCLUSION

In this paper, a hyper-heuristic improved particle swarm optimization based SVM configuration framework is proposed to resolve the big data cyber security problems. First, the SVM configuration problem is modelled as a multi-objective optimization problem with false positive rate, false negative rate and model complexity being the multiple objective parameters. This multi-objective optimization problem is solved by developing the proposed HHIPSO framework which utilizes the high-level strategy and low-level heuristics of hyper-heuristic approach and improved PSO algorithm. This proposed framework improved the selection of margin parameter, kernel type and kernel parameters for the better configuration of SVM for cyber security big data problems. The proposed framework was evaluated on two cyber security datasets: NSL-KDD and ISCX-IDS. The obtained empirical results proved that the proposed HHIPSO-SVM model provides highly superior performance compared to the other algorithms. In future, the proposed hyper-heuristic framework can be further improved by including more features of the SVM for optimization of cost function. Also, the computation time of larger datasets other than NSL-KDD and ISCX-IDS of cyber security problems will be evaluated to analyze the time complexity.

REFERENCES

1. Abomhara, M. (2015). “Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks.” *Journal of Cyber Security and Mobility*, *4*(1), 65-88.
2. Von Solms, R., & Van Niekerk, J. (2013). “From information security to cyber security.” *computers & security*, *38*, 97-102.
3. Probst, C. W., Hunker, J., Bishop, M., & Gollmann, D. (Eds.). (2010). “*Insider threats in cyber security”* (Vol. 49). Springer Science & Business Media.
4. Moore, T. (2010). “The economics of cybersecurity: Principles and policy options.” *International Journal of Critical Infrastructure Protection*, *3*(3-4), 103-117.
5. Choo, K. K. R. (2011). “The cyber threat landscape: Challenges and future research directions.” *Computers & Security*, *30*(8), 719-731.
6. Cardenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., & Sastry, S. (2009). “Challenges for securing cyber physical systems.” In *Workshop on future directions in cyber-physical systems security* (Vol. 5, No. 1).
7. Greitzer, F. L., & Frincke, D. A. (2010). “Combining traditional cyber security audit data with psychosocial data: towards predictive modeling for insider threat mitigation.” In *Insider threats in cyber security* (pp. 85-113). Springer, Boston, MA.
8. Hu, J., & Vasilakos, A. V. (2016). “Energy big data analytics and security: challenges and opportunities.” *IEEE Transactions on Smart Grid*, *7*(5), 2423-2436.
9. Babiceanu, R. F., & Seker, R. (2016). “Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook.” *Computers in Industry*, *81*, 128-137.
10. Mahmood, T., & Afzal, U. (2013). “Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools.” In *2013 2nd national conference on Information assurance (ncia)* (pp. 129-134). IEEE.
11. Kache, F., & Seuring, S. (2017). “Challenges and opportunities of digital information at the intersection of Big Data Analytics and supply chain management.” *International Journal of Operations & Production Management*, *37*(1), 10-36.
12. Sabar, N. R., Yi, X., & Song, A. (2018). “A bi-objective hyper-heuristic support vector machines for big data cyber-security.” *IEEE Access*, *6*, 10421-10431.
13. Dovom, E. M., Azmoodeh, A., Dehghantanha, A., Newton, D. E., Parizi, R. M., & Karimipour, H. (2019). “Fuzzy pattern tree for edge malware detection and categorization in IoT.” *Journal of Systems Architecture*, *97*, 1-7.
14. Shamshirband, S., & Chronopoulos, A. T. (2019). “A new malware detection system using a high performance-ELM method.” In *Proceedings of the 23rd International Database Applications & Engineering Symposium* (p. 33). ACM.
15. Zhong, W., & Gu, F. (2019). “A multi-level deep learning system for malware detection.” *Expert Systems with Applications*, *133*, 151-162.
16. Ju, A., Guo, Y., Ye, Z., Li, T., & Ma, J. (2019). “HeteMSD: A Big Data Analytics Framework for Targeted Cyber-Attacks Detection Using Heterogeneous Multisource Data.” *Security and Communication Networks*, *2019*.
17. Venkatraman, S., Alazab, M., & Vinayakumar, R. (2019). “A hybrid deep learning image-based analysis for effective malware detection.” *Journal of Information Security and Applications*, *47*, 377-389.
18. Calvert, C. L., & Khoshgoftaar, T. M. (2019). “Impact of class distribution on the detection of slow HTTP DoS attacks using Big Data.” *Journal of Big Data*, *6*(1), 67.
19. Mao, W., Cai, Z., Yang, Y., Shi, X., & Guan, X. (2018). “From big data to knowledge: A spatio-temporal approach to malware detection.” *Computers & Security*, *74*, 167-183.
20. Martín, A., Menéndez, H. D., & Camacho, D. (2017). “MOCDroid: multi-objective evolutionary classifier for Android malware detection.” *Soft Computing*, *21*(24), 7405-7415.
21. Gupta, D., & Rani, R. (2018). “Big Data Framework for Zero-Day Malware Detection.” *Cybernetics and Systems*, *49*(2), 103-121.
22. Wassermann, S., & Casas, P. (2018). “BIGMOMAL: Big Data Analytics for Mobile Malware Detection.” In *Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity* (pp. 33-39). ACM.
23. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). “A detailed analysis of the KDD CUP 99 data set.” In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1-6). IEEE.
24. Krömer, P., Platoš, J., Snášel, V., & Abraham, A. (2011). “Fuzzy classification by evolutionary algorithms.” In *2011 IEEE International Conference on Systems, Man, and Cybernetics*(pp. 313-318). IEEE.
25. Mohammadi, M., Raahemi, B., Akbari, A., & Nassersharif, B. (2012). “New class-dependent feature transformation for intrusion detection systems.” *Security and communication networks*, *5*(12), 1296-1311.

AUTHORS PROFILE

Aswanandini.R, M.Sc .,M.Phil is a Ph.D Scholar at Sri Ramakrishna College of Arts and Science, Coimbatore. She is also currently working as Assistant Professor at KG College of Arts and Science, Coimbatore. She has 8 years of Teaching experience and has published 7 papers in International Journals.

Dr.Muthumani.N is Professor & Head in the Department of Mathematics(CA) at Sri Ramakrishna College of Arts and Scienece. She has 20 years of Teaching experience and has published more than 25 papers at various International and Scopus Journals.