International Journal of Engineering Research and Science & Technology

IJERST

ISSN : 2319-5991

www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

A BI-OBJECTIVE HYPER-HEURISTIC SUPPORT VECTOR MACHINES FOR BIG DATA CYBER-SECURITY

B. VENKATESH [1], PATHIKONDA SATWIKA REDDY[2], GANDLA SAIKUMAR[3], PAMARTHI ASHOK KUMAR[4], VEMULAPALLI HARISCHANDRA PRASADU[5]

[1]Assistant professor, Dept.of CSE, Malla Reddy College of Engineering HYDERABAD.

[2,3,4,5]UG Students, Department of IOT, Malla Reddy College of Engineering HYDERABAD.

ABSTRACT:

Researchers have a formidable task in addressing cyber security concerns in the context of big data. It has been proposed that machine learning techniques may be used to deal with large data security issues. Support vector machines (SVMs) are one such approach that has shown exceptional performance on several classification issues. However, in order to set up an efficient SVM, the user must first determine the optimal SVM configuration; this is a difficult operation that calls for specialised expertise and much trial and error. In this research, we provide a formalisation of the SVM configuration procedure as a bi-objective optimisation problem, where accuracy and model complexity are treated as competing goals. We present a new problem-domain-agnostic hyper-heuristic framework for bi-objective optimisation. For the first time, a hyper-heuristic has been created specifically for this issue. High-level strategy and low-level heuristics are what make up the suggested hyper-heuristic framework. In order to determine which of many possible low-level heuristics should be utilised to produce a new SVM configuration, the high-level approach monitors search performance. The SVM configuration search space is efficiently explored by the low-level heuristics, each of which follows a unique set of rules. The proposed framework adaptively merges the benefits of decomposition- and Pareto-based techniques to approximate the Pareto set of SVM configurations, making it suitable for bi-objective optimisation. The suggested methodology has been tested on two cyber security issues: classifying Microsoft malware using a large data set and detecting anomalies in network traffic. In comparison to its contemporaries and other algorithms, the acquired results show that the suggested framework is very successful.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

SVM, Pareto, Object, and Cyber Security Issues are Keywords.

I INTRODUCTION

The advent of sophisticated technologies and the IoT in the modern age of digital information has paved the way for a massive influx of data to be produced and stored. Cyber-attacks have multiplied in tandem with the exponential expansion of online data. Cyber security solutions have been developed and used because cyber assaults inflict severe harm to networks. When it comes to protecting computers and networks against cybercrime, it's up to cyber security protocols and procedures [2]. Protecting data used in group decision-making is a top priority for these tools, as is preventing unauthorised access to networks and safeguarding sensitive data in cyberspace [3]. Many businesses rely on third-party cyber security providers like Accenture, IBM, CISCO, etc. [4] while others build their own systems in-house. In order to detect and prevent malicious behaviours, modern cyber security systems have leaned heavily on network and Internet traffic monitoring [5]. This is quite different from the standard cyber security systems that just look for forged signatures to prevent entry. While older systems attempted to identify malware by comparing incoming data to known signatures, this method is inefficient and can only uncover a subset of possible dangers [6]. Intrusion detection, firewalls, and anti-virus software are no longer effective against hackers since modern attack methodologies are much more damaging [7]. Furthermore, the advent of big data has exacerbated the precarious situation, since terabytes of data are transmitted between each node of computer networks, making it much simpler for hackers to penetrate the networks and wreak extensive damage without being detected [8]. Most of the issues with big data can be traced back to companies opening up their data networks to other parties, such as business partners and customers. This leaves the networks wide open to cyber assaults. In a similar vein, hackers' ability to bypass conventional safeguards has been bolstered by the availability of massive amounts of data. Because of big data's complexity, it's now more difficult to spot assaults while they're being launched, and the damage to hardware and software is sometimes discovered too late [9]. Big data analytics, by applying big data methods to avoid cyber-attacks, may be utilised to deal with the security risks associated with large data [10]. Many businesses have already begun updating their cyber defences in light of this idea [11]. As was previously said, machine learning algorithms have been heavily

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

used for this procedure, with the SVM coming out on top.

2. REVIEW OF TEXTS

Many studies have focused on the use of big data analytics to create effective cyber security solutions. This section covers some of the most cutting-edge methods currently available. For virus detection in large data IoT, Dovom et al. [13] introduced a fuzzy pattern tree approach. To identify malware, this technique converts the Op-codes to vector space and then uses a y fuzzy and fast fuzzy pattern tree. Results showed a high degree of accuracy in categorising, exceeding 93.13% for the Ransomware dataset and 97% for the Kaggle dataset. To identify malware, Shamshirband and Chronopoulos [14] created a high-performance ELM-based technique with a 95.72% success rate. Malware identification is aided by this paradigm, although it is limited to only three characteristics. A multi-tiered deep learning approach for identifying malware was suggested by Zhong and Gu [15]. Using a tree structure, this system organises many deep learning models, with each tree catering to the data distribution needs of a different class of malware. Despite promising experimental findings, the system's high processing time is a serious limitation for detecting malware. Targeted cyber-attacks detection using heterogeneous noisy data was suggested by Ju et al. [16]. Different types of disparate data were connected in this method to single out the bad actors. Although it is quite good at spotting cyberattacks, it only takes a handful of factors into account and ignores how people really perceive attacks. A hybrid deep learning image-based analysis approach was presented by Venkatraman et al. [17] to identify cyberattacks. This combined approach can visualise the categorization of malware and aid in the detection of questionable system behaviour. This model successfully detects malware with a high degree of accuracy and at a low computational cost. Big data sampling was utilised by Calvert and Khoshgoftaar [18] to generate different class distributions for detecting sluggish HTTP DoS assaults. This method relies on monitoring the system's legal traffic in order to identify assaults, and it uses Random forest as its learning algorithm of choice. The attack detection AUC value achieved with this method was 0.99904. However, choices are rendered statistically insignificant since only the AUC measure is evaluated. Malware detection using the big data features of cloud systems was the topic of a paper published by Mao et al. [19]. Based on the geographical and temporal characteristics of the data distributions, a graph-

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

based semi-supervised learning algorithm was developed for this method. The experimental findings demonstrated a higher malware detection rate with less computational effort. However, malware detection using file co-occurrence in end hosts has a recall upper constraint using this method. Using a multi-objective evolutionary classifier, Martin et al. [20] presented MOCDroid to identify Android malware. In this method, the malware nodes are identified by selecting groupings of import phrases using a multi-objective genetic algorithm called SPEA2. The technique is very accurate and produces few false positives, as shown by the empirical findings, but it only takes into account a limited set of goals. For detecting zero-day malware, Gupta and Rani [21] suggested a large data system based on machine learning. Attack identification is accomplished with the use of classification algorithms, and the random forests proved to be the most effective of them. In order to identify mobile malware, Wassermann and Casas [22] created the BIGMOMAL technique, which combines big data analytics with supervised-machine-learning. This method was quite effective in identifying malware in real-time, but it was plagued by the issue of notion drift. According to the research, machine learning algorithms may aid malware detection by providing more accurate categorization. It is also deduced that certain classifiers work well with just a subset of available datasets. This motivates the development of more effective configurations of machine learning algorithms, which can identify malware with greater precision while using less resources.

3. APPROACHES

The Current Structure:

SVMs are an example of a popular kind of supervised learning model used in classification and regression problems. SVMs, which are based on statistical learning theory, are superior to conventional classification algorithms in their ability to circumvent local optima. To find the best hyper plane, SVMs use a learning technique based on a kernel. In order to achieve linear separation, the kernel learning process transforms the input patterns into a higher-dimensional feature space. The current kernel functions may be broken down into two categories: local and global. While local kernel functions are effective at learning, they struggle when asked to generalise. Global kernel functions, on the other hand, are excellent at generalisation but terrible at specifics. There are both local and global kernel functions; the radial kernel function is an example of the former,

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

while the polynomial kernel function is an example of the latter. The hardest part is figuring out appropriate kernel function to utilise for a given issue instance or decision threshold. This is due to the fact that the connection between the input and output vectors (predicted variables) plays a huge role in the kernel selection process. However, in big data cyber security, the distribution of the feature space is not known in advance and may vary over the course of the solution process. since a result, the performance of the support vector machine (SVM) may be significantly influenced by the choice of kernel, since different kernel functions may prove to be effective for certain cases or at various phases of the solution process. In this study, we take a different approach by using numerous kernel functions to overcome this problem and enhance the precision of our method.

A RECOMMENDED SYSTEM

Figure 2 depicts the suggested hyper-heuristic architecture for configuration selection. The top level is the strategy, while the bottom level is the heuristics. The high-level tactic is heuristic in nature, rather than solution-based. Every time around, the high-level strategy takes a gander at the available low-level heuristics, picks one, applies it to the current solution to generate a new solution, and then determines whether or not to accept the new solution. A problem's solution space is the primary target of the low-level heuristics, which are a collection of problem-specific heuristics. We present a population-based hyper-heuristic framework that employs an archive to store the non-dominated solutions in order to solve the bi-objective optimisation issue. To efficiently estimate the Pareto set of SVM configurations, the proposed framework combines the benefits of decomposition- and Pareto (dominance-) - based techniques. We propose a method that takes advantages of both the decomposition approach's capacity for variety and the convergence capability of the dominance approach. The solutions population is the target of the decomposition method, while the archive serves as the backbone of the dominance method. Using the previous population, the archive, or both, the hyper heuristic framework creates a new population of solutions. Through this, the search is able to strike a healthy equilibrium between convergence and variety. It is important to keep in mind that although optimising for high variety entails spreading the solutions as thin as possible along PF, optimising for excellent convergence requires minimising the distances between the solutions and PF. In the sections that follow,

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

we'll talk more specifically about the key parts of the proposed hyper-heuristic framework.

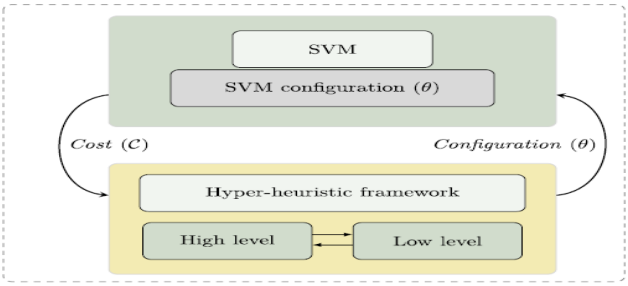
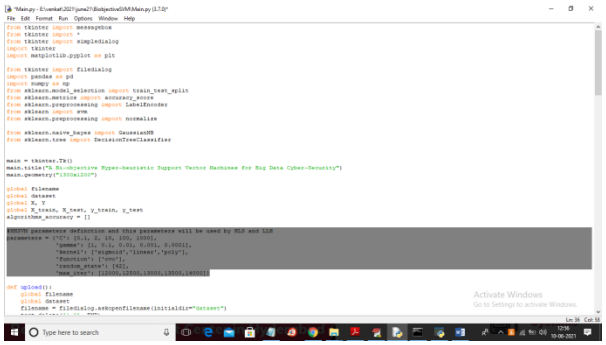


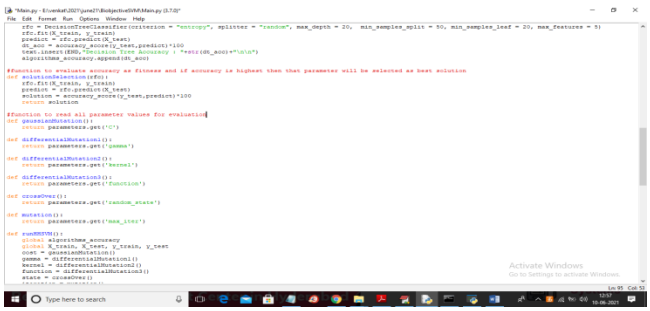
FIGURE 1. The proposed methodology.

In this study, the author explains how to use a multi-objective hyper-heuristic to choose the best possible values for the SVM optimisation parameters automatically. Optimised parameters for SVM may be chosen with the use of two hyper variables in this method, one heuristic and one strategic. The high-level tactic is heuristic in nature, rather than solution-based. The high-level approach chooses a heuristic from the available low-level heuristics, applies it to the present solution to generate a new solution, and then evaluates whether or not to accept the new solution.

First, HLH will generate a population using the chosen SVM parameters; then, LLH will apply those parameters to the population and run the SVM algorithms; if the algorithms produce the highest accuracy, LLH will accept those parameters as the solution; otherwise, the process will repeat until the accuracy is optimised. This process is repeated until the maximum fitness is attained, at which point the corresponding parameters are chosen as the optimal answer. HHSVM will determine crossover (new solution generation), mutation (new input value selection), and fitness (accuracy evaluation) at each iteration. The solution with the highest fitness value will be chosen. An example of HHSVM's implementation is shown below, annotated for your convenience.



In above screen in selected text we define parameters for HHSVM and in below screen we are using code to perform parameter selection

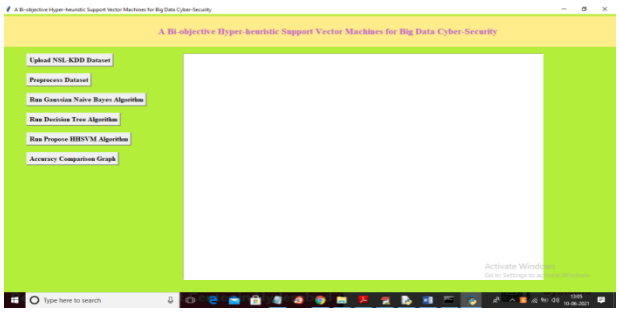


International Journal of Engineering Research and Science & Technology

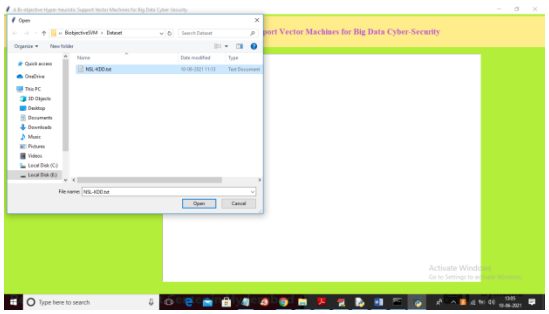
ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

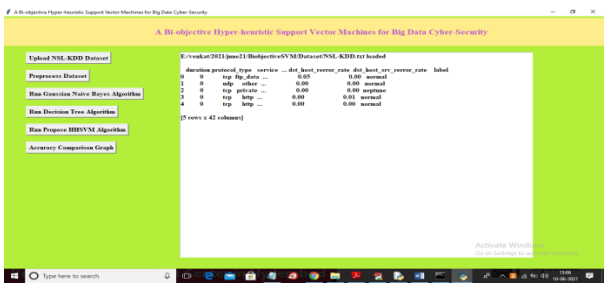
In above screen read red colour comments to select input for HHSVM and in below screen you can see HHSVM implementation



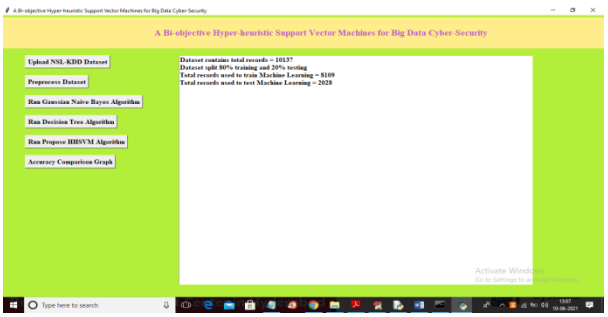
In above screen click on 'Upload NSL-KDD Dataset' button to upload dataset and to get below screen



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



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 preprocess those values to assign integer id to each unique non-numeric value

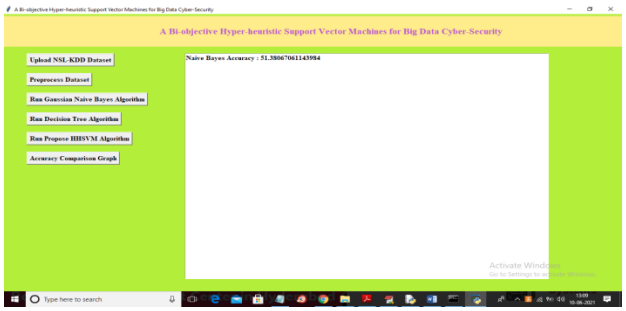


In above screen dataset is preprocessed and dataset contains huge 10137 records and application split dataset into train and test where application using 8109 records for training and 2028 records testing trained ML model accuracy. After train model test records will apply on trained model to perform prediction and then correct prediction percentage will be consider as accuracy. Now train and test dataset is ready and now click on 'Run Gaussian Naïve Bayes Algorithm' button to train Naïve Bayes with above dataset

International Journal of Engineering Research and Science & Technology

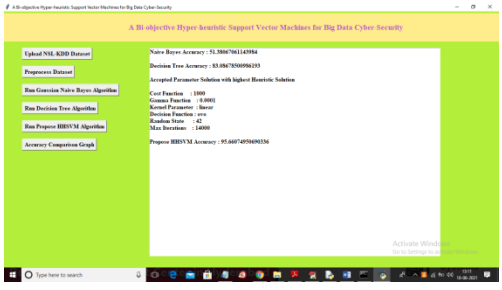
ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

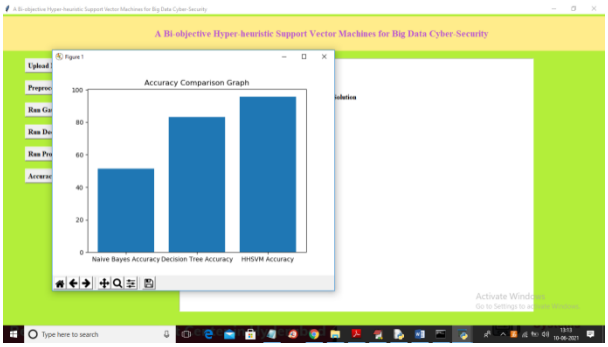


In above screen Naïve Bayes got 51% accuracy and now click on 'Run Decision Tree Algorithm' button to train decision tree on above dataset

In above screen with decision tree we got 83% accuracy and now click on 'Run Propose HHSVM Algorithm' button to train HHSVM algorithm with optimize parameters and then calculate best fitness accuracy



In above screen with optimize parameters we got 95% accuracy for HHSVM algorithm and in above accuracy line I am printing all those optimize parameters which helps in getting 95% accuracy and in above screen 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



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.

CONCLUSION

A hyper-heuristic SVM optimisation framework was presented for use with large data cyber security issues. We posed the SVM setup procedure as a bi-objective optimisation problem, with accuracy and model complexity as competing goals. The suggested hyper-heuristic framework is effective in resolving this issue of

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

bi-objective optimisation. The Pareto set of configurations is approximated by combining the benefits of decomposition- and Pareto-based techniques, both of which are included in the framework.

REFERANCES

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

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 17, Issue.2, 2024

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.