DUAL-PURPOSE HYPER-HEURISTIC SUPPORT VECTOR MACHINE DESIGNED FOR ADDRESSING CYBER SECURITY ISSUES IN BIG DATA ENVIRONMENTS

[1]Ms. G. Srividya, [2]Ch. Sridham, [3]G. Poojitha, [4]Imtiyaz Ahmad Wani
[1]Assistant Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana
[2,3,4] UG Student, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana

Corresponding Author: ***goudapollapoojitha@gmail.com***

***Abstract*— Cybersecurity within the setting of big data is known to be a basic issue and presents an incredible challenge. many new age advanced "ML" models are recommended as candidates for taking care of enormous information security issues, among those support vector machine (SVMs) have accomplished exceptional victory. However, setting up optimal SVM configurations requires high knowledge on working of SVM & cybersecurity and also manual effort. This paper introduces a dual-purposed hyper-heuristic framework that addresses the cyber security concerns in the realm of big data. The model we intend to present doesn't require manual setting of SVM configuration every time a new test data arrives.**

Keywords*— *support vector machine, hyper-heuristic, machine leaning, cybersecurity, malware detection.

I. INTRODUCTION

Analyzing and understanding big data allows researchers and organizations to make better and faster decisions, thus uplifting effectiveness of their operations.

This field has attracted the attention of researchers, practitioners, and government agencies based on big data's practical usage and challenges. However, the enormous amount of data has generated many challenges which required to be addressed.

It not only increases the scale of challenges but also produces new and different cybersecurity issues. Examples of such big data cybersecurity challenges are malware detection, authentication, and scalability issues. Among these challenges, malware discovery is a major concern. There are already numerous styles available to deal with malware discovery, such as signature-based detection methods, behavior monitoring detection methods, and pattern-based detection methods. But these pre-existing techniques are generally proposed to deal with small-scale datasets and are unfit to handle big data within a moderate quantum of time. Due to this drawback, the efficiency of them might decrease. Therefore, our team came up with a bi-objective hyper-heuristic framework for SVM configuration optimization. Hyper heuristics have more scope compared to other techniques as hyper heuristics are independent of the particular problem and continuously gain effective configurations.

Our proposed hyper-heuristic framework encompasses several key elements that separate it from existing to find an effective way to implement SVM working without the need to set several configurations (every time a new test data arrives), hence increasing the algorithm's capability and efficiency to handle large quantities of data.

II. RELATED WORK

Literature Review:
DURGESH K, SRIVASTAVA, LEKHA BHAMBHU, showcased the importance of kernels for traditional SVM to work. the paper shows the results we get while we use various types of kernels such as (Linear Kernel or Polynomial Kernel or Radial Basis Function Kernel etc.)
On a particular dataset. The result portray that choosing the appropriate kernel and appropriate values of various parameters are key for classification of a dataset.[2]
based on this paper we researched how we can improve and come up with an algorithm that performs the same as svm but doesn't require the manual setting of configurations like the kernels (Linear Kernel or Polynomial Kernel or Radial Basis Function Kernel etc.), c parameter, etc.

Yan Hou, proposed optimization of decision tree classifier. Basically, decision trees Reading and writing speed of new incoming data is slow. So to overcome this, the paper discusses "Algorithm Optimization of Decision Tree"[3] thus improving algorithms' adaptability and efficiency

Ender Özcan, Burak Bilgin, Emin Erkan Korkmaz, discussed the development and importance hyper heuristics in the field classification using machine learning algorithms. The paper gave information about the efficiency of using hyper heuristics. this paper justifies the statement that [8]"hyper-heuristics are not problem specific hence increasing the overall scope of using this for different sets of problems".

Ximing Wang & Panos M.Pardalos discussed the working of traditional svm with uncertain cases, where test data is very unique from training data. the paper discusses the loop hole in svm due to which its efficiency decreases. uncertain cases are the one that surface the limitations. [7]
Observing the conclusions of above papers we as a team decided to research and come up with an optimized solution that optimizes the working of svm .one of the papers above discussed the efficiency of the hyper heuristics, which laid the perfect base for the algorithm we have drafted.
svm's need manual setting of its configuration for it to do the task of data classification. These configurations must be set every time a new test data arrives .to avoid all this complexity we tried to use hyper heuristic approach. hyper-heuristics are not problem specific solutions, hence from the pool of solutions our proposed algorithm selects one solution and then chooses weather to accept result this particular solution gives. The way this hyper heuristic is implemented is discussed in methodology.

III. METHODOLOGY

The proposed framework has two levels:

The proposed framework has levels in it. One is called the high-position strategy .Other one is called as low-position heuristics.. In each replication, the high-position strategy selects one from the being pool of low-position heuristic, applies it to the Present result, to come up with a updated result and also makes a decision on whether to accept this new result. The low positions heuristics are a collection of problem-specific heuristic that try to provide solutions to specific set of problems. We try to apply these two-position framework differently as explained below:

As our research is majorly focused on malware detection, we here try develop a framework similar to svm. The proposed framework not only performs all activities as svm does but also overcomes two major drawbacks of a support vector machine.

Here is the list of drawbacks of SVM our proposed framework overcomes:

- SVM can't just work directly on data used for training and give an accurate classification of test data for it to work setting certain SVM configuration (c-parameter, kernel function, etc.) is mandatory, these parameters can be given only by individuals having expertise in the "ML" and "Cybersecurity".
- SVM can either give speedy result or faster result. (we are incorporating both hence our proposed solution is "bi objective")

Our proposed model gives speedy and accurate results without any manual setting of SVM configuration. In our proposed method we are trying to develop a framework this would include developing website using Django framework This would include developing a website using the Django framework (python language) and my SQL.

Data flow and working:

our website would allow users to enter details such as "what kind of problem the user's website is undergoing (such as got hacked, lost or stolen data from the website, etc). Based on user input our algorithm will classify the issues into various groups and inform what kind of virus users' websites might be subjected to (such as polymorphic virus or macro virus).

here we are doing the same work as SVM which would be the classification of data here our algorithms group training data into various groups each titled with a unique virus name.

whenever a user inputs his website issues, based on input the algorithm will decide which group the input issues (given by the user) belong to, hence providing the user the type of virus the user's website must have undergone. here our algorithm does similar work as work svm which is classification and detection but our algorithm gives speedy and accurate results without the need to set configurations frequently (which might be necessary if we are using svm).

The following block diagrams showcase the data flow. we are trying to build these block diagrams to present the way through which data is been gathered from the user and classify effectively.

User side data flow (Fig 1): this depicts how user accesses the website and how they enter the information. The data entered by user is the sole input for the developed algorithm. There are various fields of information that must be entered by user/client

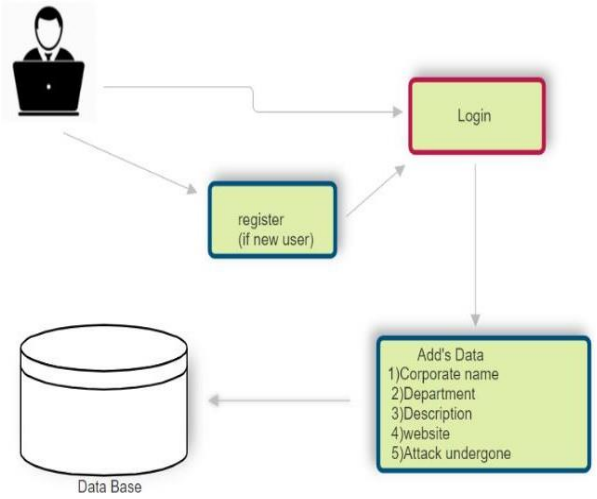

Fig 1: user side data flow

Algorithms internal working (Fig 2): it shows how the proposed algorithm (which is referred as "modified svm" in diagram) accesses the clients input data and then choses to which data cluster (made of training data) that new client input might belong to after clustering stores data in database in organized way .as we working on cyber security training data would be viruses name and their impact based on clients input (about their website) algorithm decides which virus must be associated to user input. then it stores all of this in database

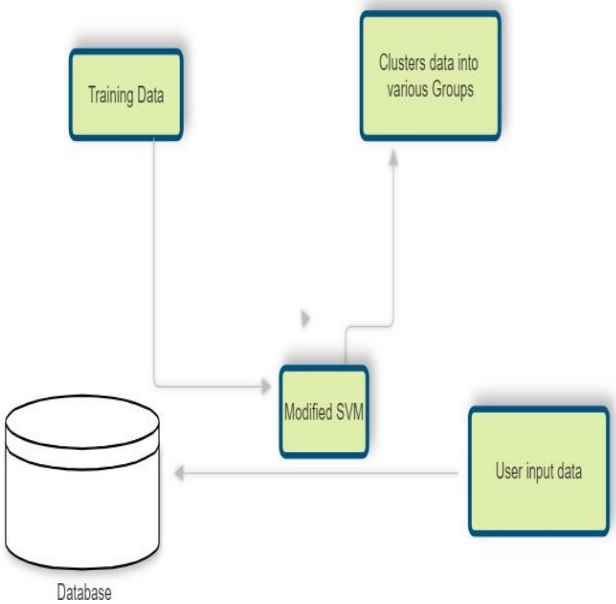

Fig 2: algorithm's internal working

IV. RESULTS AND DISCUSSIONS

- **GRAPHS**

Here the fig 3 clearly depicts and shows comparison of existing algorithms and proposed algorithm based of metrics (accuracy, precision and recall). These three metrics showcase how good an Algorithm perform a classification task and also some miscellaneous detection tasks. Data set which is provided for module to train. There are certain formulas to calculate these metrics which are later discussed in the paper. On a general basis the algorithm with high accuracy, recall and precision is considered to be the most efficient at classifying the data. We can clearly see the proposed algorithm (which is hyper heuristic svm) has way more accuracy, recall and precision compared to existing classification algorithms (which are random forest, decision tree classifier and traditional svm). Here traditional svm refers to svm which require manual setting of svm configuration.

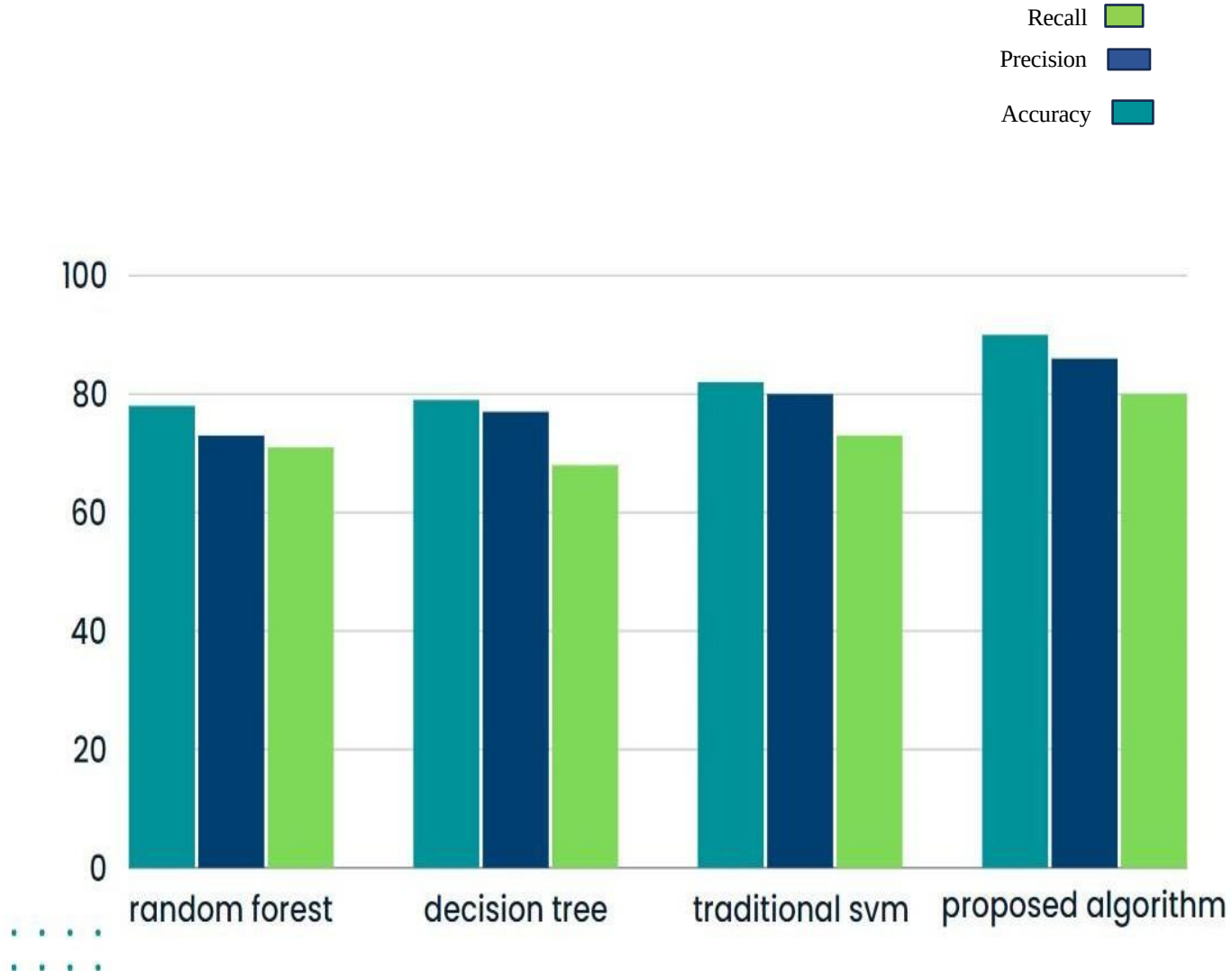

fig 3: Algorithms performance comparison graph

- **TABLES**

Table 1(value comparison of algorithms): showcases the values of accuracy precision and recall. Accuracy, precision and recall are metrics utilized to check the overall performance of an algorithm

Formulas:

Accuracy=Number of Correct Predictions/Total Number of Predictions

Precision=True Positives + False Positives/True Positives

Recall=True Positives/True Positives + False Negatives

Using these formulas the values of accuracy, precision and recall are calculated. The values required for formulas only be found if we train the algorithm with certain data set. We have first collected some existing algorithms that perform data classification. these are the one already discussed in the above graph section. Our proposed algorithm is referred as hyper-heuristic support vector machine. Now we trained all these algorithms on same data set by which we were able to quantify the accuracy precision and recall of each algorithm.by observing the values in table, we say that proposed algorithm has more (accuracy, precision and recall) compared to existing algorithm. As we know that these metrics indicate efficiency of an algorithm, conclusion of proposed algorithm being more efficient at classifying the big data compared to existing algorithms can be surfaced.

Training data utilized in the calculation of these values is dataset well suited for the developed project.as discussed in methodology were trying to build a website that takes user input (user website name, attack undergone etc.) the algorithm takes user input as test data. training data includes list of various names of viruses along with attributes associated with it(as we are trying to build malware detection mechanism)

Table 1 : **value comparison of Algorithms**

ALGORITHMS	Accuracy	Precision	Recall
RANDOM FOREST	78.65	73.13	71.014
DECISION TREE CLASSIFIER	79.77	77.04	68.11
TRADITIONAL SVM	82.02	80.3	73.14
Modified HYPER HEURISTIC SVM (PROPOSED ALGORITHM)	90.03	86.1	80.46

V. CONCLUSION

This study, a hyper-heuristic SVM optimization model is suggested to face with cybersecurity issues. We remodeled the SVM setup process as a dual- goal optimization issue, where precision and model intricacy are seen as opposing priorities. This dual-goal optimization issue can be dealt through the hyper-heuristic system. It merges the capabilities of decomposition- and Pareto-based methods.

VI. REFERENCES

[1] Yingchun Liu, proposed an optimized upgradation of Random forest algorithm in big data environment, COMPUTER MODELLING & NEW TECHNOLOGIES 2014.

[2] DURGESH K, SRIVASTAVA, LEKHA BHAMBHU, discussed on “DATA CLASSIFICATION USING SVM", Journal of Theoretical and Applied Information Technology,2010.

[3] Yan Hou, proposed optimized approach of Decision Tree Algorithm for Big Data Analysis,Advances in Intelligent Systems Research,International Conference on Transportation & Logistics, Information & Communication,2018.

[4] Malak El Bakry,Soha Safwat,Osman Hegazy, researched and published paper on "Big Data Classification using Fuzzy KNN",2015.

[5] Edmund K Burke, Michel Gendreau, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan & Rong Qu provided an analysis on topic

“Hyper-heuristics: a survey of the state of the art”, Journal of the Operational Research Society ,(2013)

[6] Cobos C, Mendoza M and Leon E (2011) put forward A hyper-heuristic approach to design and tuning heuristic methods for web document clustering. In: Evolutionary Computation (CEC). IEEE: New Orleans, LA, pp 1350–1358.

[7] Ximing Wang & Panos M. Pardalos,performed a Survey of Support Vector Machines with Uncertainties, Annals of Data Science,2015

[8] Ender Özcan,Burak Bilgin,Emin Erkan Korkmaz,did a comprehensive analysis of hyper-heuristic,Annual International Conference on Information and Sciences (AiCIS),Article in Intelligent Data Analysis