MINI PROJECT REPORT

SVM Hyperparameter Tuning Using Harris Hawk Optimisation

Submitted in partial fulfilment of the requirements for the degree of

Bachelor of Engineering
in
Computer Science and Business System

Submitted By

Triambak Sharma - 101918006
Tanya Srivastava - 101918031
Ishita Mullick - 101918035

Under the supervision of:

Dr. Prashant S Rana
Associate Professor,

Computer Science & Engineering Department,

Thapar Institute of Engineering & Technology, Patiala

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

December 2021

Table of Contents

1. Introduction

Over the years, technology has revolutionised the way we live, and today, data drives the world. It became imperative to analyse this data, and this led to the advent of concepts such as Machine Learning and Artificial Intelligence. It has now evolved to a stage where we are able to train algorithms using historical data and predict the future.

Various studies have revealed that the prediction power of ML and AI algorithms can be improved by optimising their parameters. Therefore, we present a comprehensive view of how these machine learning algorithms can be made more efficient by using nature-based optimisation techniques.

1.1 Nature Based Optimisation Algorithm – Harris Hawk Optimisation

In the past few years, many nature-based metaheuristic algorithms such as Particle Swarm Optimisation and Firefly algorithms have come up, which are becoming increasingly popular due to their high efficiency. These algorithms are inspired by nature and produce acceptable solutions to complex problems with high efficiency.

The Harris Hawk algorithm is one such metaheuristic that mimics the hunting behaviour of Harris Hawks and was first introduced by Heidari et al. (2019). The Harris Hawk is different than other predatory raptors as they attack in a unique cooperative foraging manner with other family members living in the same stable group, while others carry out this process alone.

In this strategy, several hawks attack strategically from different directions on a detected prey (usually a rabbit) that is trying to escape. These cooperative tactics are extremely advantageous as they lead to the exhaustion of the prey and make it more vulnerable and defenceless. It can, therefore, not escape from this besiege.

There are two phases in the common search phase in all conventional optimization methods - exploration and exploitation. Exploration helps to generate new and varied responses, while exploitation focuses on searching the local region by using the information of a current good solution found in the region

Now, we model the exploratory and exploitative phases of the HHO that have been inspired by the attacking strategies of Harris Hawks. Fig. 1. illustrates all of HHO's stages, which are discussed in the subsections below..

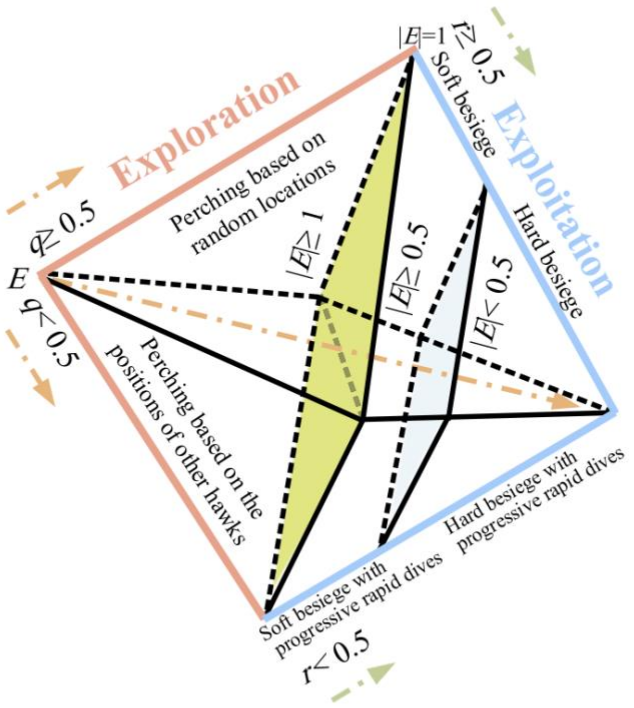


Fig. 1. Different phases of HHO

1.1.1 Exploration Phase

We first describe the exploration mechanism. In HHO, the hawks perch on some random locations and wait to detect the prey. In case the hawks are not able to track and detect the prey, they wait, observe, and monitor the site to detect the prey. The position of each of the hawks is a candidate solution and at each step, the best candidate solution is considered as the prey or optimum.

1.1.2 Transition from exploration to exploitation

The algorithm's transition from exploration to exploitation is determined by the prey's fleeing energy. The energy of the prey decreases after each attack.

1.1.3 Exploitation phase

The Harris Hawks make their notorious surprise pounce during the exploitation phase. They go for the prey that was discovered in the previous phase. They employ four assault tactics to effectively pursue the victim. Depending on how much energy the prey has left, the hawks will either use a hard or a gentle besiege to grab it. To maximise their chances of a cooperative kill, the hawks strive to approach the target from diverse directions. The prey will continue to lose energy, and the hawks' hunt for the prey will increase. As a result, we switch between the gentle and hard besiege procedures using the escaping energy E.

Let the chance of the prey to successfully escape is *r*, and if r < 0.5, the prey escapes successfully after the surprise pounce. When |E| ≥0.5, the soft besiege happens, and when |E| <0.5, the hard besiege occurs. When |E| ≥0.5 but r < 0.5, soft besiege with progressive rapid dives occurs i.e. a soft besiege is constructed before the surprise pounce. The Levy Flight (LF) concept is used to mimic the zigzag deceptive motions of the rabbit and the consequent irregular, abrupt, and rapid dives of hawks. Levy Flight activities are the optimal searching tactics for predators in non-destructive foraging conditions.

When |E| <0.5 and r < 0.5, that the rabbit doesn't have enough energy to escape, and in this case, a hard besiege is conducted before the final pounce to kill the prey. The hawks go closer to the prey and decrease the distance of their average location.

1.2 Support Vector Machine (SVM)

A Support Vector Machine is a supervised machine learning algorithm. SVM supports high generalization and offers better accuracy and faster prediction compared to its counterparts. It is a robust and computationally efficient algorithm. The SVM algorithm's main goal is to categorise data points individually by finding a hyperplane in an N-dimensional space, where N is the number of features. This objective boils down to an optimization problem as the algorithm works its way through the data points to find the most optimal hyperplane.

The regularization parameter, C, is utilized to regulate the trade-off between the maximum of margin and several misclassifications in SVM. Second, nonlinear SVM kernel functions are employed to translate training data from a low-dimensional input space to a higher-dimensional feature space.

A penalty term is added which helps to balance out between the trade-off between increasing the distance of decision boundary to the support vectors and or maximising the number of data points that can be correctly classified.

C parameter adds a penalty for each misclassified point. The case in which C is small, the penalty levied for each point that is misclassified is low. This results in selection of a decision boundary having a large margin at the expense of a greater number of misclassifications. On the other hand, if C is large the penalty is high for any misclassification and hence SVM tries to minimize any misclassification resulting in a smaller margin. It is noteworthy that the penalty is not constant for all misclassified points is proportional to the distance of the point from the decision boundary.

The gamma parameter determines how far a training point's impact extends. A smaller gamma value suggests a higher similarity radius, which means a larger number of points are clustered together.

When gamma is high, the points must be quite near in order to be clustered together. When the gamma value is large, the model is frequently overfitted. Following figures represent the effects of gamma.

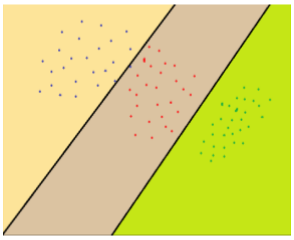


Fig. 6. Low Gamma

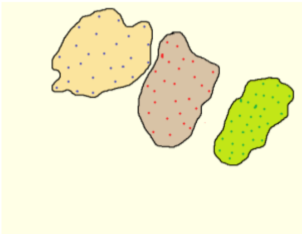


Fig. 7. High Gamma

In Fig. 6., similarity radius is very large resulting in all the points present in the coloured region to be considered in the same class. On the contrary in Fig. 7. the data points need to be very near to each other and fall in a tightly bounded area to be classified in a single group. A small noise may result in a data point being misclassified further making it difficult for the model to be generalised. Small values of gamma make the regions separating different classes more generalised. Large gamma may nullify the effect of C and small gamma leads to C affecting the model similar to case of linear model. Typically values of C range between (0.1,10) and gamma between (0.0001,10), although specific optimal values differ based on the application.

1.3 The proposed HHO-SVM approach

We have produced a hybrid algorithm by combining the Support Vector Machine Algorithm with the nature-based Harris Hawk Optimization Algorithm (HHO). Through this approach, we have tried to optimize the accuracy of the SVM classifier by determining the values of the parameters of the SVM model using HHO. It has been described in greater detail in the methodology section.

2. Background

Many application fields effectively apply metaheuristics methods, including classification and feature optimization (i.e., feature selection and parameter optimization). Studies reveal that these algorithms improve the prediction power of ML and AI algorithms by optimising their parameters.

Metaheuristic algorithms have also found their use in Epilepsy detection (Asmaa Hamad, 2018). The research proposed a hybrid EEG classification approach by using Gray Wolf Optimizer (GWO) to enhance the support vector machines (SVMs) for automatic seizure detection. The GWO optimisation technique was used to select the optimal parameters of SVM and significant feature subset to obtain a successful EEG classification. In a study by Lamiaa M. el Bakrawy, 2017, researchers used the hybrid algorithm of Gray Wolf Optimiser and Naïve Bayes classifier for heart disease diagnosis.

In the domain of civil engineering, these algorithms have contributed by showing that the solution to the problem of soil stability lies in adjusting computational weights of their four conditioning factors. The research (Hossein Moayedi et al, 2019) presented a new optimisation of the ANN and HHO optimisation for stability analysis of soil slopes. It was demonstrated that synthesizing the HHO algorithm can effectively assist ANN in learning and predicting the slope failure pattern.

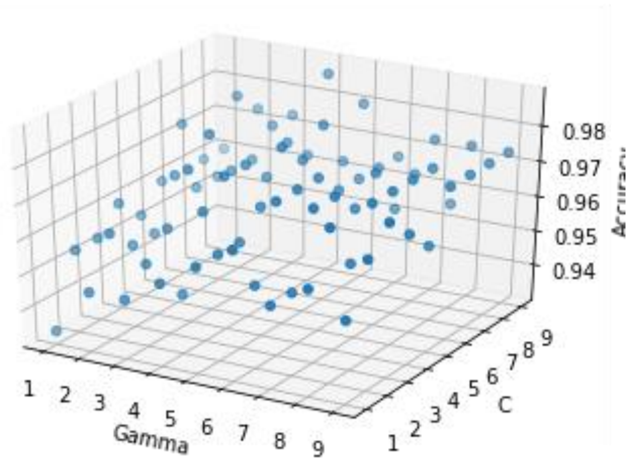
HHO has also proven to be a better optimisation technique that other optimsation techniques including artificial bee colony (ABC), wind driven optimization (WDO) and atom search optimization (ASO) algorithms (Erdal Eker et al, 2019). From the statistical analysis carried out, the optimization of the HHO algorithm had a better performance.

The Particle Swarm Organisation technique has been used to improve accuracy of the estimation process based on the SVM technique when enough training samples were not available. This study (Yakoub Bazi et al, 2007) concentrated on situations in which gathering a statistically significant and representative number of training samples was difficult and time-consuming. Their experiments confirmed that the optimisation process obtained by combining a metaheuristic algorithm is more reliable and more adapted to tackle the challenging complexity of the semi supervised regression problem.

By reading many research papers on concepts that integrated metaheuristic algorithms and ML algorithms, we realised the scope of nature-based optimisation techniques, and hence decided to integrate it with SVM and analyse their efficiency by using them on various datasets.

3. Objectives

The machinery and technology around us are highly inspired by nature and its works. The academic curriculum so far has exposed us to various domains of computer science and mathematics ranging from statistics and probability to Machine Learning etc. With this project we try to delve deeper into the domain of optimization giving specific attention to the nature based techniques and how they can be applied to tune hyperparameters of Machine learning models. The main objective is to understand how HHO works and how it can be integrated with Machine Learning so as to tune the hyperparameters of SVM . Secondly, we try to draw a contrast between nature based optimization technique and a randomized approach and understand how efficient is nature based optimization when it comes to tuning of hyperparameters.



The above plot depicts that the relationship between C, gamma and the accuracy is non deterministic. The value of hyperparameters are solely subjective to the type of dataset that the model is being trained for. It can be clearly seen that there isn’t much correlation among the variables and nature based optimization can be used to tune the parameters.

Although nature based optimization is a vast domain and highly researched area, but during our research a typical issue was faced. Major research was concentrated on the MATLAB platform, making it difficult for us to proceed with our research due to our limitation of usage of the platform and our relatively higher familiarity with python. We realised that this is something which may be a problem faced by many and how it may be hindering the exploitation of these wonderful nature inspired techniques by python users. Therefore we chose to rewrite the HHO code from scratch on the python platform to make it available to a wider audience and also benefit from the vast libraries and open source capability of the language.

4. Methodology

The following libraries were used during the code build up.

1. Pandas
2. Numpy
3. Sklearn
    a. sklearn.svm for SVC function
    b. sklearn.model_selection for train test split, Kfold and cross_val_Score
    c. sklearn.metrics for accuracy score
4. matplotlib.pyplot
5. mpl_toolkits.mplot3d for Axes3D
6. Random
7. Math
8. Warning

The default RBF kernel was used during training of Support Vector Machine, Label Encoder was used for pre-processing. Below is the flow chart explaining the workflow of HHO-SVM.

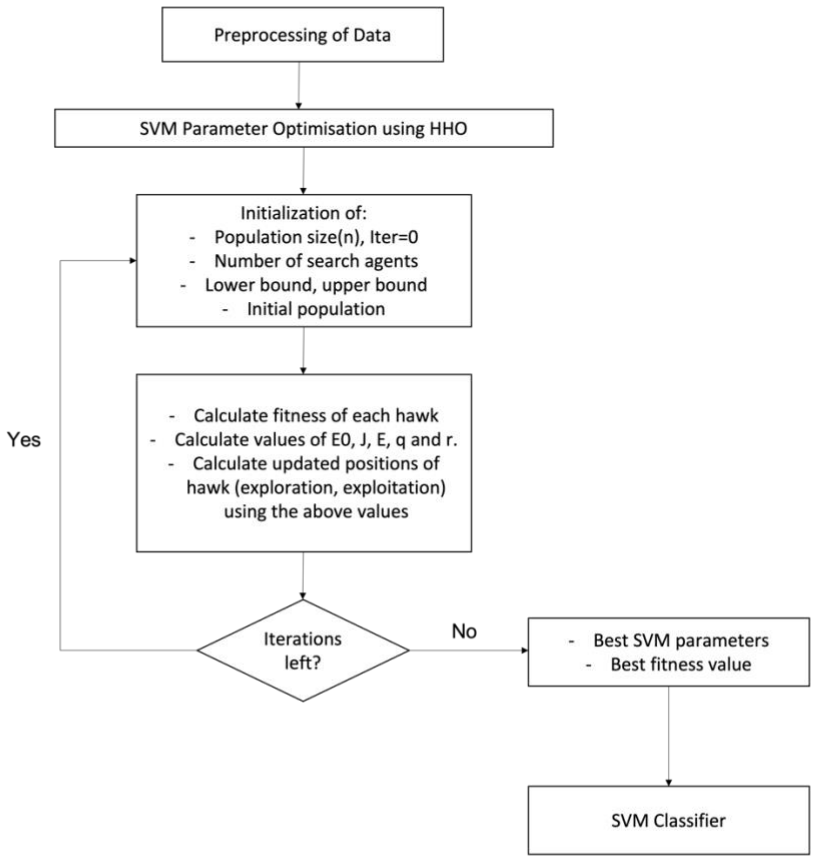


Fig. 8. Proposed HHO-SVM Flowchart

For categorization problems, swarm intelligence algorithms mixed with machine learning approaches are currently used. Hence, we have integrated the Harris Hawk Optimisation technique with the SVM algorithm. The HHO-SVM approach has 3 phases - (1) Pre-processing, (2) Parameter Optimisation, and (3) Classification and validation.

The following is the pseudo-code of the SVM and HHO algorithm -

```
while (number of iterations) do
    Bring the population in range
    Find the average position of the current population of hawks (Xm)
    Set Xrabbit as the location of rabbit (the hawk that gave maximum accuracy)
    for (every hawk) do
      Calculate initial energy E0 and jump strength J,
      E0 is a random value between -1 and 1,
      J = 2*(1-random())
      Further, use E0 to calculate E = 2 * E0 * (1 - t/T)

      if (|E|>=1)   then                          #Exploration phase
            Find random value of probability of perching strategy q
            if (q>=0.5) then                      #First perching strategy
              Update location vector using Eq. (1).
            else                                  #Second perching strategy
              Update location vector using Eq. (1).
            endif
      else                                        #Exploitation phase
            r = random.random()
            if r>=0.5 and |E|>=0.5:   then              #Soft Besiege
              Update location vector using Eq. (4).

            else if r>=0.5 and |E|<0.5:  then           #Hard Besiege
              Update location vector using Eq. (6).

            else if r<0.5 and |E|>=0.5:   then      #Soft Besiege with rapid dives
              Update location vector using Eqs. (8), (9), (10) and (11).
```

Call SVM

else if r<0.5 and |E|<0.5: **then** #Hard Besiege with rapid dives

Update location vector using Eqs. (2), (11), (12) and (13).

Call SVM

end if

end if

end for

Check if current fitness obtained is greater than optimal fitness

Accordingly, update optimal fitness and global optimal fitness

Increment iteration

end while

5. Observations and Findings

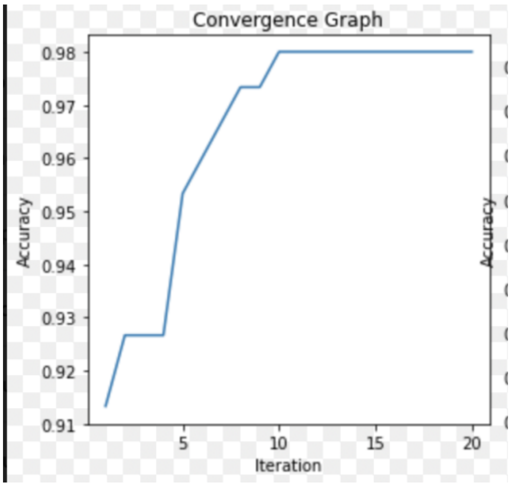
The HHO-SVM and SVM model were trained for the following datasets.

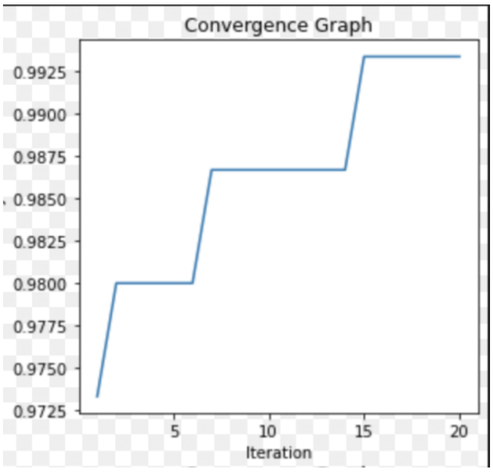
1. Iris Dataset
2. Pima Indian Diabetes Database
3. Hepatitis C Dataset
4. HR Analytics
5. Campus Placement Dataset

Convergence graphs for the above datasets during runtime are attached below.

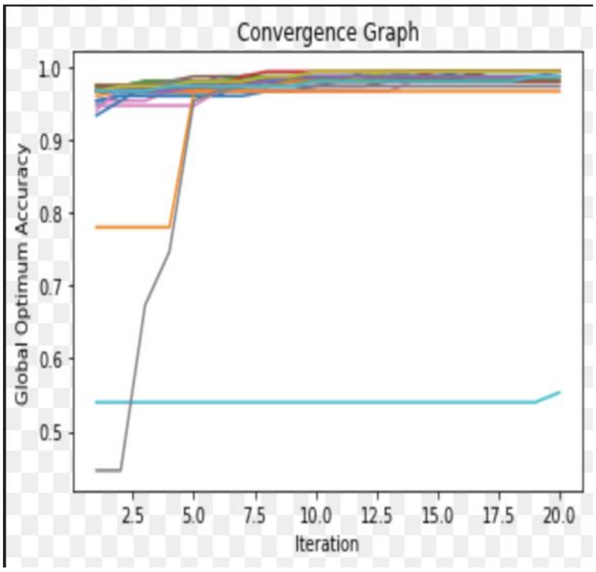
1. Iris Dataset

- Convergence Graph for 2 individual runs



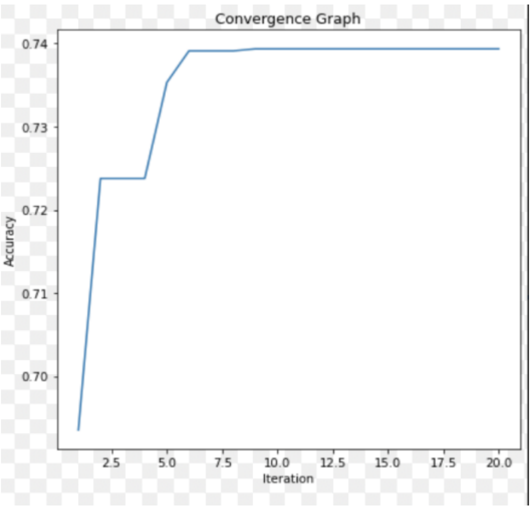


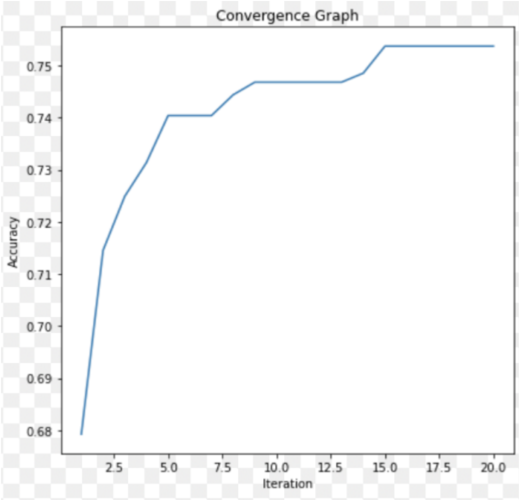
- Convergence graph for all 30 runs



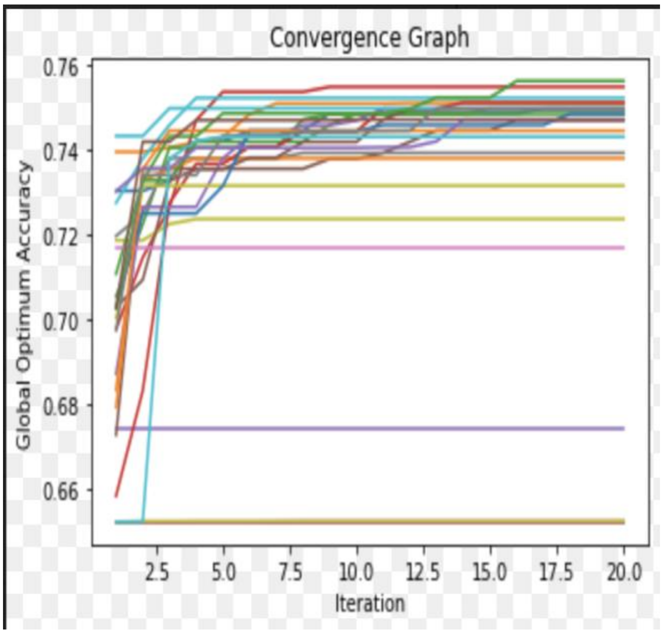
2. Pima Indian Diabetes Database

- Convergence Graph for 2 individual runs



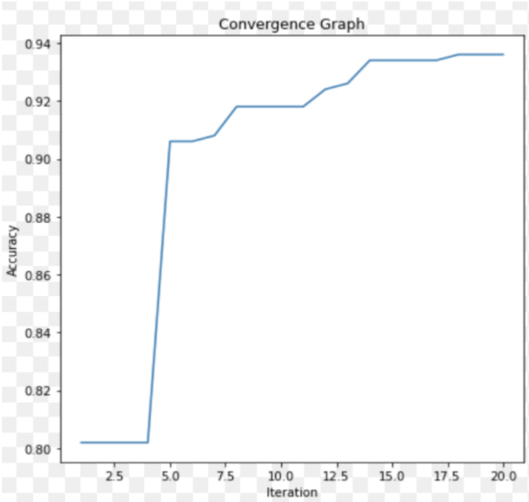


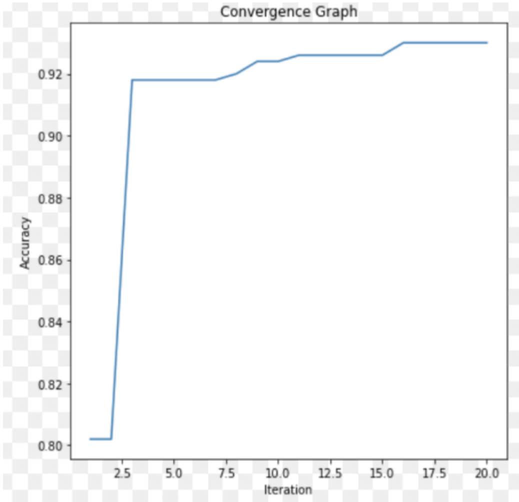
- Convergence graph for all 30 runs



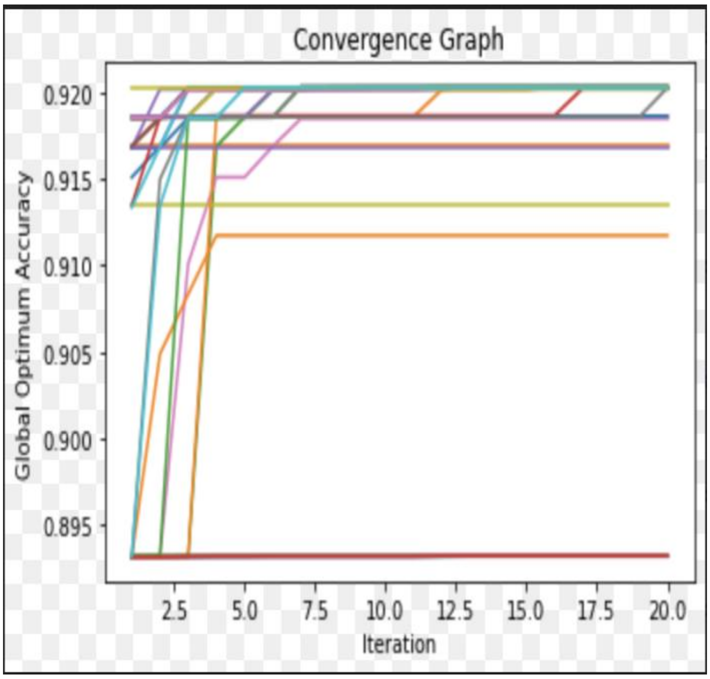
3. Hepatitis C Dataset

- Convergence Graph for 2 individual runs



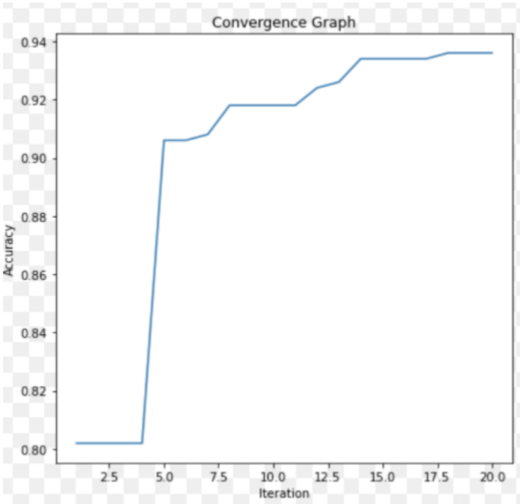


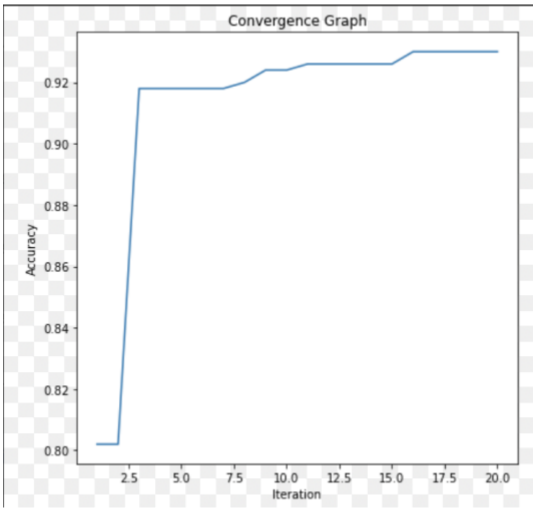
- Convergence graph for all 30 runs



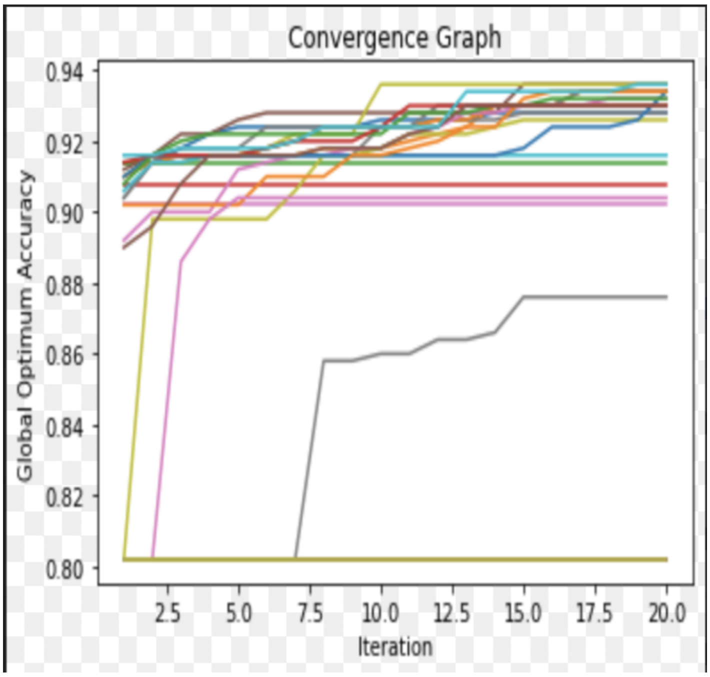
4. HR Analytics

- Convergence Graph for 2 individual runs



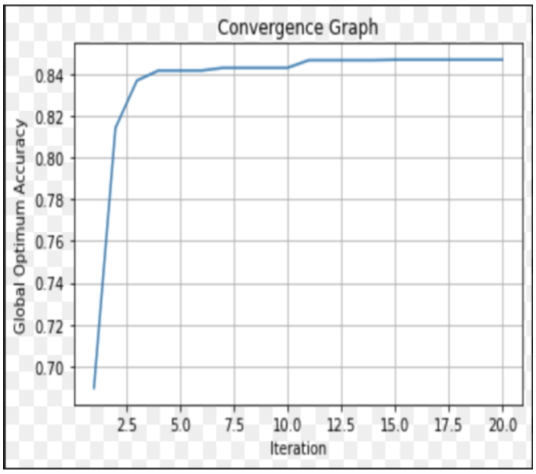


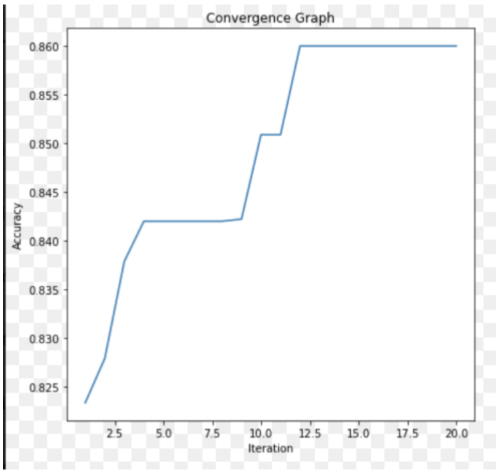
- Convergence graph for all 30 runs



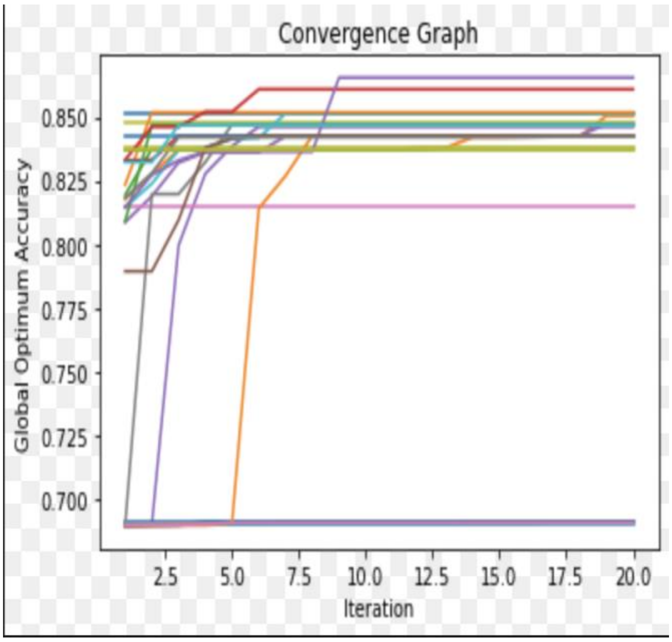
5. Campus Placement Dataset

- Convergence Graph for 2 individual runs





- Convergence Graph for 2 individual runs

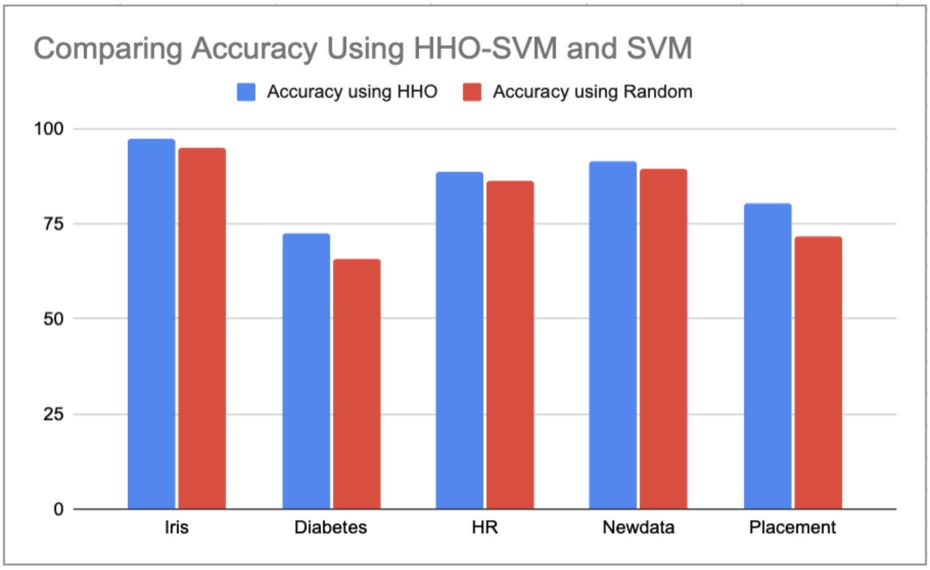


Both models (SVM, SVM-HHO) were trained and 30 iterations were performed for each dataset on both models.

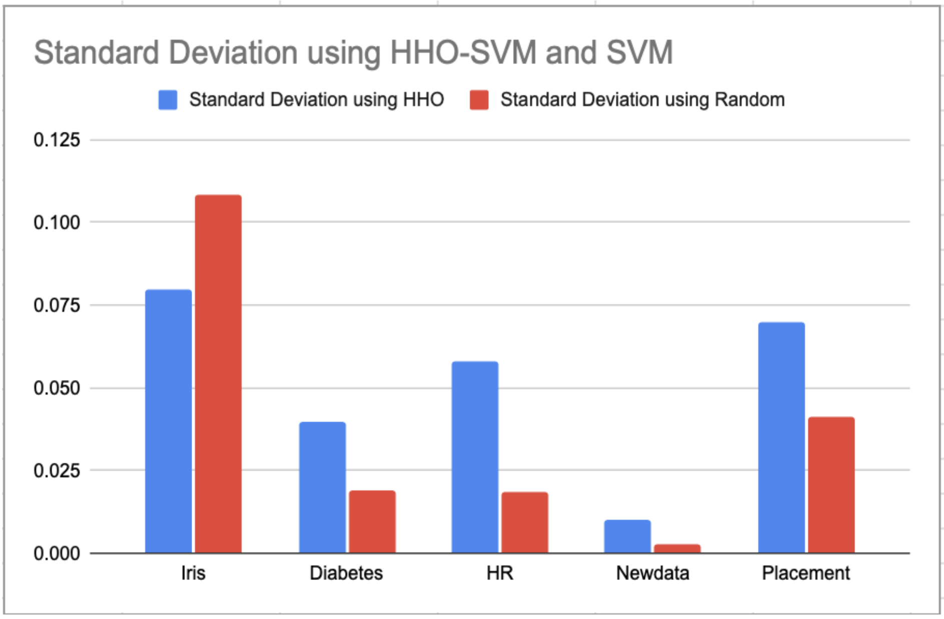
Below is a snippet of the obtained result for Campus Recruitment Dataset.

Run Number	Accuracy (HHO_SVM)	Accuracy (random)
1	0.8515151515	0.8363636364
2	0.8504329004	0.69004329
3	0.8374458874	0.7597402597
4	0.6906926407	0.6896103896
5	0.8461038961	0.6896103896
6	0.6906926407	0.69004329
7	0.6906926407	0.6896103896
8	0.6911255411	0.6971861472
9	0.8478354978	0.7192640693
10	0.8512987013	0.725974026
11	0.6909090909	0.7112554113
12	0.8517316017	0.7443722944
13	0.8374458874	0.6896103896
14	0.8424242424	0.6891774892
15	0.8471861472	0.8095238095
16	0.6906926407	0.6963203463
17	0.8145021645	0.7199134199
18	0.846969697	0.8233766234
19	0.8376623377	0.7305194805
20	0.6904761905	0.7212121212
21	0.8426406926	0.745021645
22	0.8426406926	0.7352813853
23	0.8467532468	0.6976190476
24	0.8608225108	0.6896103896
25	0.8653679654	0.6898268398
26	0.8426406926	0.6893939394
27	0.6906926407	0.7064935065
28	0.8422077922	0.6893939394
29	0.8374458874	0.6898268398
30	0.8467532468	0.6891774892
Average	0.8038600289	0.7181457431
standard deviation	0.06987929493	0.04117567843

The average accuracy and standard deviation of the thirty runs were computed for each dataset and the following graphs were plotted to interpret the results and draw contrast between HHO-SVM and SVM.



The conclusion from the above graph is that HHO-SVM outperforms the SVM without the nature base optimization. Also, major differences can be seen in the Placement database where notably the accuracy of HHO-SVM is approximately 80% as compared to the SVM without the nature based optimization being approximately 71%.



The conclusion from the above graph is that accuracy observed in the 30 runs deviate vastly among each other in HHO-SVM as compared to SVM without nature based optimization. This gives a surprising observation that even though the HHO-SVM has high standard deviation among the obtained accuracies, still the overall accuracy of HHO-SVM is better.

6. Limitations

While developing the project idea, we realised that this project had some limitations –

1. The agenda at hand in itself was very unique as we realized there was a dearth of previous research in the field of parameter tuning of ML algorithms using nature based algorithm. While we found a lot of research using nature based optimisation techniques in feature selection, not many had tried to integrate them, especially HHO, for prediction modelling.

2. Due to nature based algorithms being a highly specialized and mathematical concept, we couldn't come across much course curriculum which made it difficult to handle such a the concept especially since we were trying to integrate it with SVM.

3. Since this is a highly specialised topic, there is not much material to refer to online, and not many people are aware of the existence of this concept. Thus, everything had to. Be studied and read from the grass root level and research papers, most of which were quite complex and took the help of limited resources to understand.

4. All existing codes for nature based optimisation exist only in MATLAB. Hence, we first had to gain a deep understanding of MATLAB concepts and the concept of the Harris Hawk optimisation technique. Only then were we able to successfully translate the code into Python.

5. In the initial runs, we had the problem of pooling of values at the boundaries. Since we had to ensure that our data lied between the lower bound and upper bound, we had to bring the data within the range after every calculation. However, due to issues such as the type of dataset, and some other minor problems in the algorithm, most of the values were getting revalued to either the upper bound or the lower bound. By tweaking the algorithm and applying more complex datasets, we were able to obtain the desired result which also proved that the algorithm was very efficient even when the datasets were very complex.

6. Another problem we observed in the application of the algorithm was that there was not enough variation in the parameters of the Support Vector Machine parameters - gamma and c. This was because of the a package we were using – the scikit library. It was using an

internal optimizer because of which it was already setting the values of the parameters and decreased the efficiency of the Harris Hawk Optimiser. We solved this problem by using the K(10) cross validation algorithm on our dataset.

7. Running the algorithm on the datasets was a task as it was very time consuming. Each run took a minimum of 15 minutes in the linear kernel, for larger datasets, it took even longer. Due to this, testing and debugging became more difficult. We tried to change kernels, but even when we used the RBF kernel, it took more than 10 minutes to run 30 iterations. This was perhaps one of the biggest limitations of using an this integrated algorithm as it made us bound by limited processing power.

8. Since this project is very specialised, there are no codes available on the integration of HHO with SVM, and as a result, we had to develop the entire logic ourselves. In case we faced any issue, such as the issues we described above, we had to delve deep into the limited resources available and figure it out ourselves. We did not receive many solutions to problems such as pooling of values at the boundary and only got a solution after a lot of trial-and-errors.

9. HHO has a huge mathematical background which takes time to understand. Understanding how and why the 2 phases occur, as well as when and how the four types of attacks occurred took a lot of time and effort. Even then, the mathematics was difficult to figure and we struggled while understanding the logic. There were concepts such as Levy Flight that were an important part of the HHO algorithm, which increased its complexity.

10. We tried looking into discretization to try and solve the problem of pooling of values at the boundary but didn’t get good results. After going through various research papers, we came to the conclusion that the reason could be the distribution of the continuous values not being uniform.

7. Conclusions and Future Work

The above project work led to the conclusion that HHO-SVM performs better than the typical SVM. The project has resulted in a deeper understanding of the HHO technique and SVM algorithm, and also enlightened us about various other nature-based optimization techniques learnt throughout our project work. We have also come to an understanding that there is much more to explore about both nature-based optimization and the internal working of various Machine learning algorithms. During the project work, we encountered various other limitations which went out of the current scope of our knowledge base and also tested our time constraints. These limitations majorly include our limited mathematical knowledge which was required at a level we didn’t estimate In the current project we limited ourselves to only Harris Hawk Optimization integration with Support Vector Machines and contrasted its evaluated results with normal SVM i.e without nature-based optimization. In the future, we hope to scale this project further onto different Machine learning algorithms and also explore more towards integration of different nature-based optimization techniques. We propose to contrast results among not only the same model but different models also. We further would likely work upon contrasting nature-based optimized parameters with other optimization techniques to make our finding more robust and generalized.

8. Bibliography/References

[1] E. Eker, M. Kayri and S. Ekinci, "Investigation of Performance of HHO Algorithm in Solving Global Optimization Problems," 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), 2019, pp. 1-6, doi: 10.1109/IDAP.2019.8875909.

[2] Y. Bazi and F. Melgani, "Semisupervised PSO-SVM Regression for Biophysical Parameter Estimation," in IEEE Transactions on Geoscience and Remote Sensing, vol. 45, no. 6, pp. 1887-1895, June 2007, doi: 10.1109/TGRS.2007.895845.

[3] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, Dušan Fister, A Brief Review of Nature-Inspired Algorithms for Optimization, 2013;

[4] A. Hamad, E.H. Houssein, A.E. Hassanien, A.A. Fahmy, A hybrid eeg signals classification approach based on grey wolf optimizer enhanced SVMs for epileptic detection, in International Conference on Advanced Intelligent Systems and Informatics (Springer, 2017), pp. 108–117

[5] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris Hawks optimization: Algorithm and applications. Future Gener Comput Syst 2019; 97:849–72.

[6] Moayedi, H., Abdullahi, M.M., Nguyen, H. et al. Comparison of dragonfly algorithm and Harris hawks optimization evolutionary data mining techniques for the assessment of bearing capacity of footings over two-layer foundation soils. Engineering with Computers 37, 437–447 (2021).

[7] El Bakrawy LM. Grey Wolf Optimization and Naive Bayes classifier Incorporation for Heart Disease Diagnosis. Australian Journal of Basic and Applied Sciences. 2017 May;11(7):64-70.

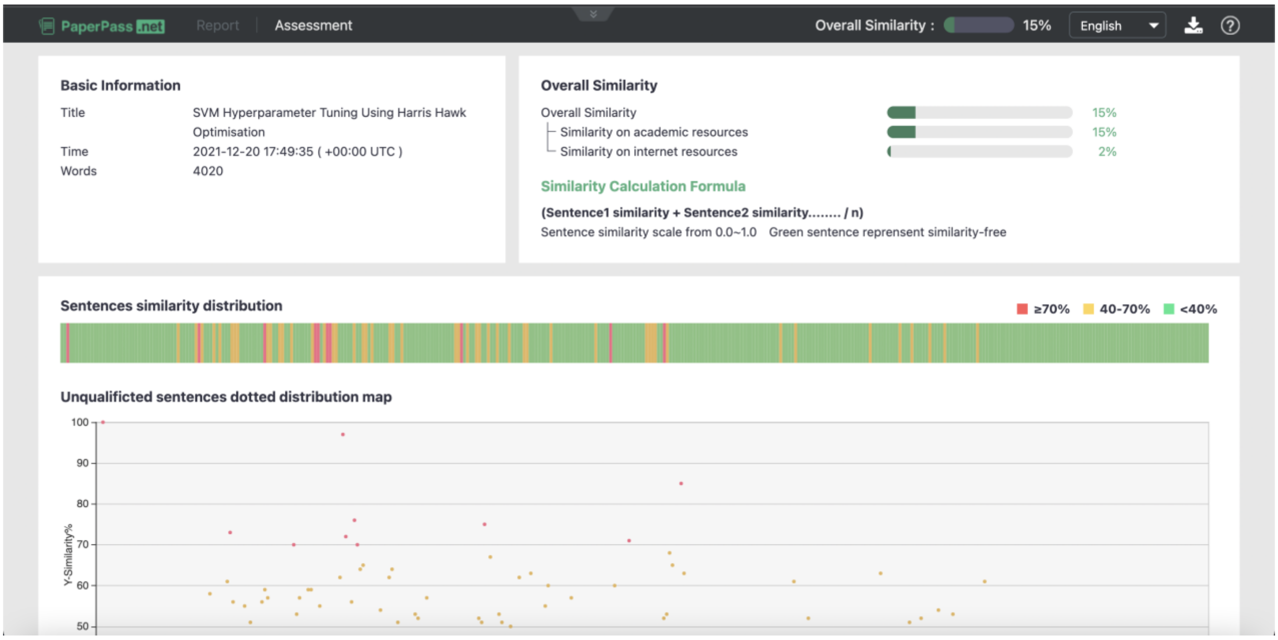
[8] Moayedi, H., Osouli, A., Nguyen, H. and Rashid, A.S.A., 2019. A novel Harris hawks' optimization and k-fold cross-validation predicting slope stability. Engineering with Computers, pp.1-11.

[9] Wenlong Fu, Jiawen Tan, Xiaoyuan Zhang, Tie Chen, Kai Wang, "Blind Parameter Identification of MAR Model and Mutation Hybrid GWO-SCA Optimized SVM for Fault Diagnosis of Rotating Machinery", Complexity, vol. 2019, Article ID 3264969, 17 pages, 2019.

[10] Abbasi, A., Firouzi, B. & Sendur, P. On the application of Harris hawks optimization (HHO) algorithm to the design of microchannel heat sinks. Engineering with Computers 37, 1409–1428 (2021).

[11] Malik, A., Tikhamarine, Y., Sammen, S.S. et al. Prediction of meteorological drought by using hybrid support vector regression optimized with HHO versus PSO algorithms. Environ Sci Pollut Res 28, 39139–39158 (2021).

9. Plagiarism Report



Submitted To -

Dr. Prashant S Rana

Submitted By –

(Triambak Sharma)

(Tanya Srivastava)

(Ishita Mullick)