Supplementary Material of "Enhanced Feature Based Granular Ball Twin Support Vector Machine"

A. Quadir[1], M. Sajid[1], M. Tanveer[1]*, and P. N. Suganthan[2]

[1] Indian Institute of Technology Indore, Simrol, Indore, India
{mscphd2207141002,phd2101241003,mtanveer}@iiti.ac.in
[2] KINDI Center for Computing Research, College of Engineering, Qatar University, Doha, Qatar
{p.n.suganthan}@qu.edu.qa

S.I Time complexity and algorithm of the proposed EF-GBTSVM model

The complexity of the proposed EF-GBTSVM model mainly hinges on three factors: (a) the computation of granular balls, (b) the necessity of matrix multiplication to generate the hidden feature matrix, and (c) the use of TSVM for classification. The time complexity of standard TSVM [2] is $\mathcal{O}(\frac{n^3}{4})$. Our approach begins with the training dataset $\mathcal{X}$, which we consider as the initial granular ball (GB) set. We initially divide this GB into two granular balls using the 2-means clustering method, with a time complexity of $\mathcal{O}(2n)$. In subsequent phases, if both granular balls are impure, they are further divided into four granular balls, each maintaining a maximum time complexity of $\mathcal{O}(2n)$. This iterative process continues for a total of *iter* iterations. Therefore, the overall time complexity of generating granular balls is $\mathcal{O}(iter \times 2n)$ or less, accounting for the maximum time complexity per iteration and the total number of iterations *iter*. Generating the hidden feature matrix H involves multiplying the generated granular ball center matrix by randomly generated weights, with a time complexity of $\mathcal{O}(k^2h)$. Hence, the overall time complexity of the proposed EF-GBTSVM is (or less than) $\mathcal{O}(\frac{k^3}{4}) + \mathcal{O}(iter \times 2n) + \mathcal{O}(k^2h)$, where k represents the total number of generated granular balls and h represents the number of hidden nodes. The detailed algorithm of the proposed EF-GBTSVM model, as outlined in 1.

S.II Comparison of the proposed EF-GBTSVM model w.r.t. the baseline GBSVM and TSVM models

This section outlines the difference between the proposed EF-GBTSVM model and existing GBSVM and TSVM models.

- EF-GBTSVM vs GBSVM

* Corresponding Author

Algorithm 1 Algorithm of the proposed EF-GBTSVM model.

Input: Purity threshold η, and the traning dataset $\mathcal{X}$.
Output: Model parameters.

1: Assume the entire dataset $\mathcal{X}$ is represented as a granular ball GB and set of granular balls, G, to be empty set, i.e., $GB = T$ and $G = \{\ \}$.
2: $Temp = \{GB\}$.
3: **for** $i = 1 : |Temp|$ **do**
4:     **if** $pur(GB_i) < \eta$ **then**
5:         Split GB_i into GB_{i1} and GB_{i2}, using 2-means clustering algorithm.
6:         $Temp \leftarrow GB_{i1}, GB_{i2}$.
7:     **else**
8:         Compute the center $c_i = \frac{1}{p}\sum_{j=1}^{p} x_j$ of GB_i, where $x_j \in GB_i$, $j = 1, 2, \ldots, p$, and p is the number of training sample in GB_i.
9:         Compute the label t_i of GB_i, where t_i is assigned the label of majority class samples within GB_i.
10:         Put $GB_i = \{(c_i, t_i)\}$ in G.
11:     **end if**
12: **end for**
13: **if** $Temp \neq \{\ \}$ **then**
14:     Go to step 3 (for further splitting).
15: **end if**
16: Set $G = \{GB_i,\ \ i = 1, 2, \ldots, k\} = \{(c_i, t_i),\ \ i = 1, 2, \ldots, k\}$, where c_i signifies the center of the granular ball, t_i is the label of GB_i and k is the number of generated granular balls.
17: Find the hidden layer features using (4).
18: Create the enhanced features using (5).
19: Compute w_1, b_1, w_2, and b_2 using (18) and (19).
20: Classify testing samples into class $+1$ or -1 using (22).

- The proposed EF-GBTSVM model solves two quadratic programming problems (QPPs) to determine the optimal parameters. However, GB-SVM solves one large QPP to obtain the optimal hyperplanes, leading to an increase in time complexity compared to the proposed models.
- The proposed models utilize the external package "CVXOPT" to solve the dual of the QPPs, employing the "qp-solvers" function to obtain the global solution, whereas, GBSVM employs the PSO algorithm (an iterative method), which may converge to local minima rather than the global minimum.

– EF-GBTSVM vs TSVM

- The proposed model's effectiveness is attributed to its utilization of granular balls as inputs rather than individual sample points. This allows the EF-GBTSVM to efficiently handle large datasets and demonstrate scalability. However, the TSVM's imperative demand for matrix inversions presents formidable obstacles to its efficiency and applicability on large-scale datasets.

- TSVM struggles to handle noise and outliers in datasets, whereas the proposed model addresses these challenges effectively by incorporating granular balls to generate optimal classifiers.

S.III Sensitivity Analyses

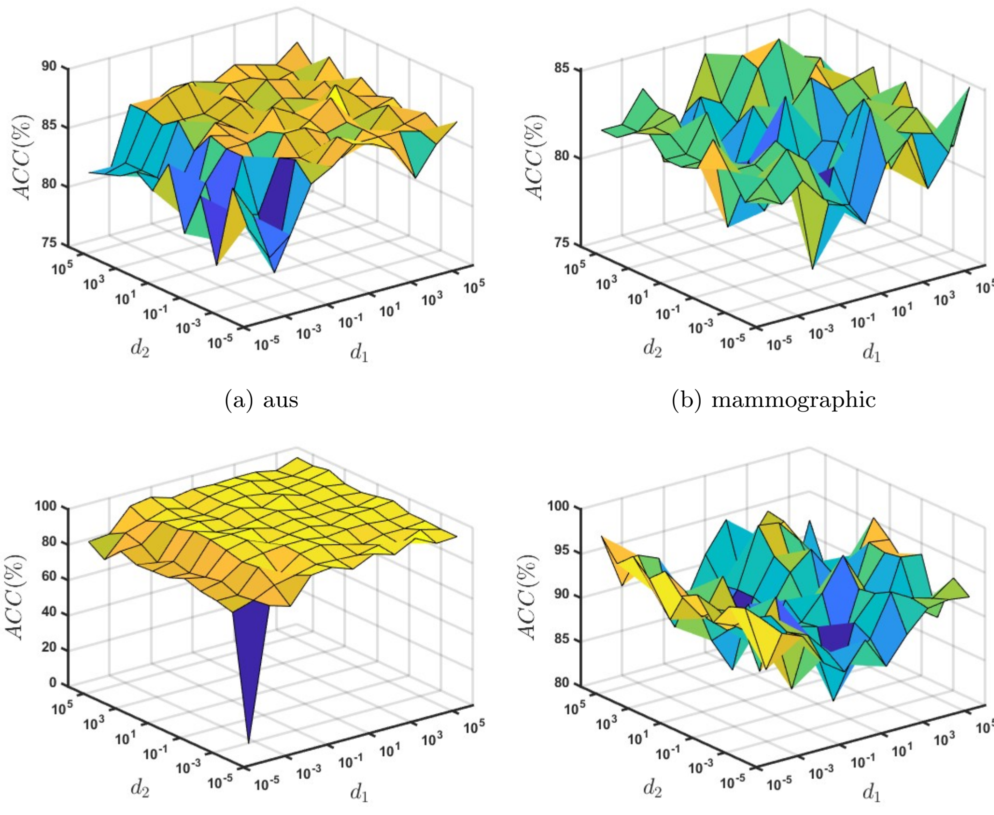
To comprehensively understand the robustness of the proposed models, it is essential to analyze their sensitivity to hyperparameters. Therefore, we conduct the following sensitivity analyses to delve deeper into the behavior of the models:

S.III.A Sensitivity Analysis of Hyperparameters d_1 and d_2

To thoroughly grasp the nuanced effects of hyperparameters on the model's generalization capability, we systematically explore the hyperparameter space by varying the values of d_1 and d_2. This exploration allows us to identify configurations that maximize predictive accuracy and enhance the model's robustness to unseen data. The graphical representations in Fig 1 offer visual insights into how parameter tuning affects the accuracy (ACC) of our EF-GBTSVM model in the linear case. These visuals illustrate significant variations in model accuracy across different d_1 and d_2 values, underscoring the sensitivity of our model's performance to these hyperparameters. From Figs. 1a and 1b, the optimal performance of the proposed EF-GBTSVM model is observed within the d_1 and d_2 ranges of 10^{-1} to 10^5 and 10^{-3} to 10^5, respectively. From Figs. 1c and 1d, the ACC of the proposed EF-GBTSVM model archives the maximum when d_1 and d_2 ranges of 10^{-3} to 10^5, respectively. Therefore, we recommend using d_1 and d_2 from the range 10^{-3} to 10^5 for efficient results, although fine-tuning may be necessary depending on the dataset's characteristics for the proposed EF-GBTSVM model to achieve optimal generalization performance.

S.III.B Effect of Parameter "Act fun" on the Performance of the Proposed EF-GBTSVM model

The activation function significantly influences the performance of the EF-GBTSVM model. In our experiment, we tuned nine different activation functions. The indexing of these functions is as follows: 1) SELU, 2) ReLU, 3) Sigmoid, 4) Sine, 5) Hardlim, 6) Tribas, 7) Radbas, 8) Sign, and 9) Leaky ReLU. We investigate the relationship using Fig 2 across datasets including breast_cancer, chess_krvkp, haber, and heart-stat. Our observation highlights a sensitivity to "Act fun". For instance, in the breast_cancer dataset, activation function 9 shows lower performance. Conversely, in the haber dataset, activation function 4 exhibits superior performance. This variability underscores the mixed performance across different datasets with respect to Act fun, suggesting the importance of fine-tuning the activation function to optimize results effectively.



(a) aus

(b) mammographic

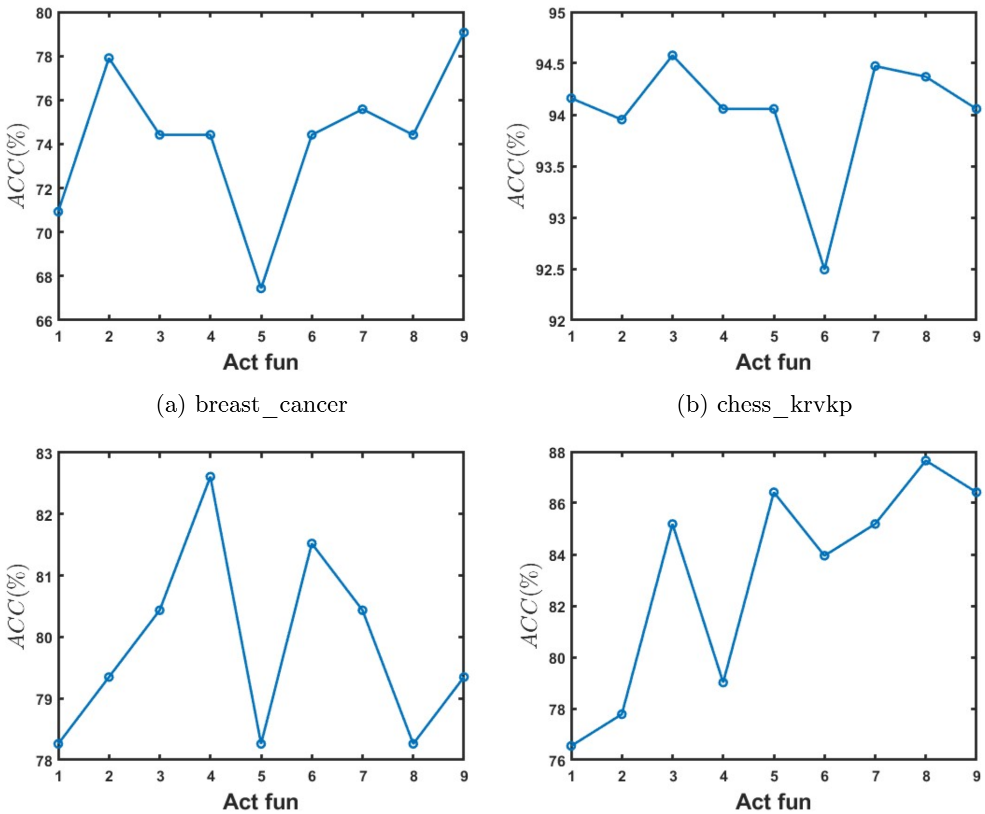
(c) monks_3

(d) yeast-0-2-5-7-9_vs_3-6-8

Fig. S.1: The effect of hyperparameter (d_1, d_2) tuning on the accuracy (ACC) of some UCI and KEEL datasets on the performance of EF-GBTSVM.

S.III.C Influence of the numbers of hidden modes h

The impact of hyperparameter h (numbers of hidden nodes) is illustrated in Fig 3. Our analysis reveals distinct trends for the EF-GBTSVM model. For thyroid dataset, the performance shows steady improvement with increasing h, plateauing at higher values like $h = 143$ or greater. Conversely, EF-GBTSVM for checkerboard_Data achieves peak performance at $h = 103$ and exhibits a gradual decline as h increases beyond this point. In conclusion, our results underscore the impact of dataset characteristics on the performance of the proposed EF-GBTSVM model, stressing the importance of fine-tuning the parameter h for optimal performance of the proposed EF-GBTSVM model.



(a) breast_cancer (b) chess_krvkp

(c) haber (d) heart-stat

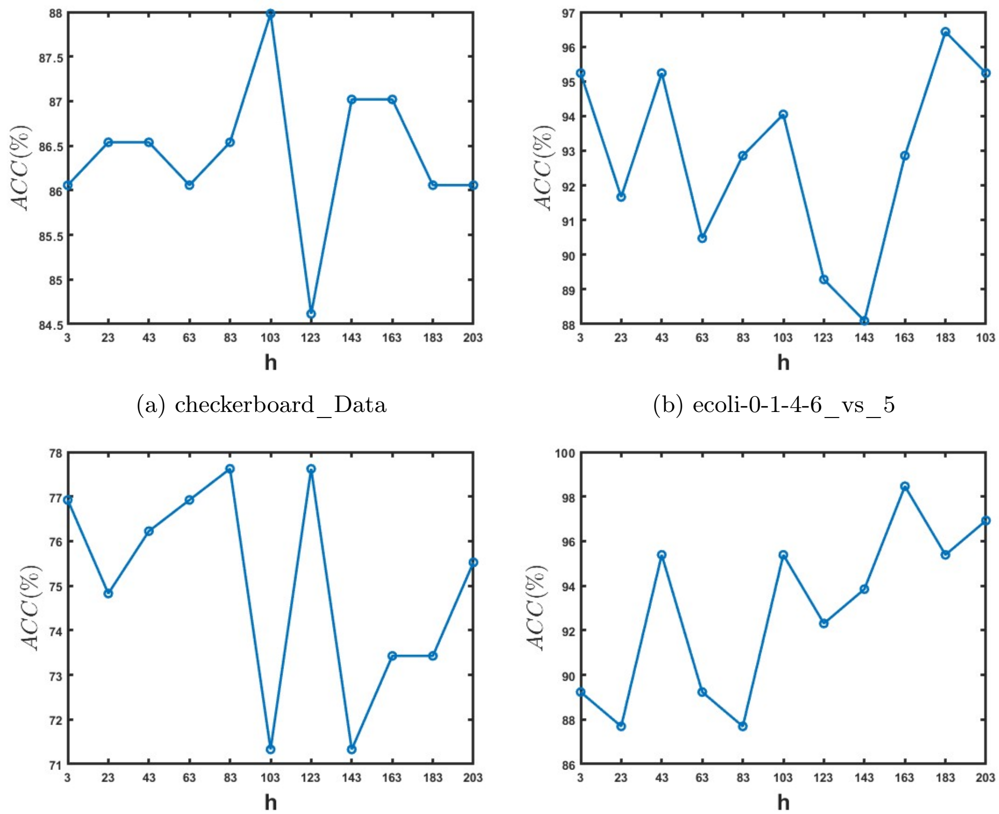
Fig. S.2: Effect of parameter "Act fun" on the performance of the proposed EF-GBTSVM model.

Dataset	TSVM [2]	RVFLwoDL [1]	RVFL [3]	HF-TSVM	HF-GBTSVM	EF-TSVM	EF-GBTSVM
aus	64.31	87.98	86.94	83.75	85.1	87.02	88.96
checkerboard_Data	64.31	85.98	85.94	43.75	85.1	87.02	86.21
chess_krvkp	67.41	90.2	90.41	69.04	90.09	97.5	98.16
haber	57.96	76.09	78.26	73	76.09	76.09	79.35
monks_3	59.7	43.11	43.11	82.608	88.62	93.41	92.22
Average ACC	62.74	76.67	76.93	70.43	85	88.21	88.98

Table S.1: Ablation study of the granular ball in the proposed EF-GBTSVM model compared with baseline TSVM, RVFLwoDL, and RVFL models over UCI and KEEL datasets.

S.IV Ablation Study

We conducted an ablation study on the EF-GBTSVM model to confirm the importance of granular balls in improving model performance. In this study, we



(a) checkerboard_Data (b) ecoli-0-1-4-6_vs_5

(c) musk_1 (d) thyroid1

Fig. S.3: Effect of parameter h on the performance of the proposed EF-GBTSVM model.

compared the EF-GBTSVM model's performance against baseline models, including TSVM, RVFLwoDL, and RVFL. Additionally, we trained HF-GBTSVM and EF-GBTSVM models using the original input samples instead of the generated granular ball centers, referring to these as HF-TSVM and EF-TSVM models, respectively.

From the results shown in Table 1, the EF-GBTSVM model demonstrated superior performance across most datasets, achieving an average accuracy (ACC) of 88.98%, the highest among the models compared. The EF-TSVM model followed closely with an average ACC of 88.21%. This indicates that using granular balls, along with extracting features in the enhanced feature space, is crucial for enhancing model performance.

The ablation study highlights the significance of granular balls in the EF-GBTSVM model. By comparing the models trained with and without granular ball centers, we observed a clear performance advantage when incorporating granular balls. This suggests that the granular ball approach not only helps in capturing the underlying data structure but also improves the robustness and ACC of the model. The study underscores the critical role of both the granular ball framework and feature extraction in the enhanced feature space in achieving superior classification results.

Bibliography

[1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[2] Jayadeva, R. Khemchandani, and S. Chandra. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):905–910, 2007.

[3] Y.-H. Pao, G.-H. Park, and D. J. Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2): 163–180, 1994.