

Optimizing SVM Hyper parameters for Imbalanced Datasets using Decision Boundary Visualization

¹Wong W K and ²Leena B

¹Department of Electrical and Computer Engineering, Curtin University Malaysia, 98009 Miri, Sarawak, Malaysia.

²Department of Electronics and Communication Engineering, KGISL Institute of Technology, Tamil Nadu, India.

¹weikitt.w@ieee.org, ²leena.bojaraj@kgkite.ac.in

Correspondence

Leena B
leena.bojaraj@kgkite.ac.in

Article Info

Journal of Future Networks and Communications
(<https://sepub.tw/journals/jfnc/jfnc.html>)

©2025 The Author(s).
Published by SE Publications.

Received 18 November 2024
Revised from 20 December 2024
Accepted 30 December 2024
Available online 05 January 2025

This is an open access article under the CC BY-NC-ND license. (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Abstract – Support Vector Machines (SVM) are commonly used for binary classification tasks, but their performance often fails in the presence of imbalanced datasets. The classifier tends to be inclined towards the common class, leading to suboptimal decision boundaries. This research investigates the impact of hyper parameter optimization on the performance of SVM when applied to imbalanced datasets, with a focus on visualizing decision boundaries. The main contribution of this research is the introduction of a dynamic class weighting strategy combined with an automated hyper parameter tuning approach, which adapts to class imbalances. The metrics considered in this research includes the class weights, which is optimized using grid search and cross-validation techniques. Additionally, decision boundary visualization is used to qualitatively assess the impact of different hyper parameter settings on the classifier's performance. Experiments are conducted using synthetic datasets with varying levels of class imbalance, and the results are evaluated based on classification accuracy, precision, recall, and the area under the ROC curve (AUC). The results reveal that using dynamic class weights and optimizing hyper parameters significantly improved classifier performance, with a 15% increase in recall and a 10% improvement in AUC, compared to standard SVM approaches.

Keywords – Binary Classification, Imbalanced Datasets, Decision Boundary, ROC Curve, Accuracy.

1. INTRODUCTION

The performance of Support Vector Machines (SVM) in high-dimensional spaces and their capacity to produce strong decision boundaries have made them one of the most well-liked and extensively applied methods for binary classification issues. When it comes to standard binary classification, SVM seeks to minimize classification errors and maximize the margin between two classes. SVM, on the other hand, has a tendency to favour the majority class when datasets are unbalanced, meaning that one class has a larger number of cases than the other. Because of this bias, the classifier's ability to correctly predict the minority class is negatively impacted by inadequate decision limits [1].

In many real-world applications, such as fraud detection, medical diagnosis, and anomaly identification, where the minority class reflects the uncommon but crucial events or outcomes, imbalanced datasets are a typical problem. Standard SVM classifiers frequently attain high accuracy in these situations by categorizing the majority class of samples, but this leads to poor performance for the minority class. Numerous strategies, including class weighting, cost-sensitive learning, and resampling, have been proposed to address this problem. Among these, dynamic class weighting has shown promising results by directly adjusting the decision function to give more importance to the minority class [2, 3].

This study aims to investigate the impact of hyper parameter optimization on SVM performance when applied to imbalanced datasets, specifically by introducing a dynamic class weighting strategy combined with automated hyper parameter tuning. The optimization of hyper parameters such as the regularization parameter (C), kernel type, and class weights can significantly affect the performance of the SVM, particularly in imbalanced classification tasks. Additionally, this research emphasizes the importance of visualizing decision boundaries to qualitatively assess how different hyper parameter settings impact the classifier's decision-making process. By optimizing these hyper parameters and analyzing

the resulting decision boundaries, we aim to provide insights into how to enhance SVM's ability to handle imbalanced data and improve performance metrics such as recall, precision, and AUC [3].

The paper will begin with an introduction to SVM and the challenges of classifying imbalanced datasets. It will then review related work on SVM hyperparameter optimization and decision boundary visualization. The methodology section will cover dataset preparation, SVM model configuration, and the hyperparameter optimization process, followed by decision boundary visualization techniques. Results will discuss the impact of hyperparameter changes on model performance and decision boundaries, concluding with key findings and suggestions for future work.

2. RELATED WORKS

The issue of imbalanced datasets has been extensively studied in the machine learning community. Several approaches have been proposed to address the challenges posed by class imbalance, particularly in SVM-based classifiers. One of the most widely adopted methods is class weighting, where different weights are assigned to the classes based on their frequency in the dataset. C-SVM, for instance, allows the use of a `class_weight` parameter to adjust the importance of the minority class during the optimization process. Class weight tuning has been found to improve classifier performance, particularly in terms of recall and precision for the minority class, although it may sometimes reduce overall accuracy due to the shift in focus towards the minority class [4].

Another technique used in imbalanced classification is re-sampling. While these methods can improve classifier performance, they may introduce issues such as overfitting (in the case of oversampling) or loss of information (in the case of under sampling). Some studies have explored the combination of re-sampling techniques with SVMs to improve their generalization ability while avoiding bias towards the majority class [5]. Kernel selection is another important aspect of SVMs that impacts their performance in imbalanced settings. While linear kernels are often used due to their simplicity and efficiency, non-linear kernels such as the Radial Basis Function (RBF) kernel have been shown to be effective in handling complex, non-linearly separable data. The kernel can significantly influence the decision boundary and the classifier's ability to handle imbalances, as non-linear kernels are better suited for capturing the underlying patterns in imbalanced datasets [6].

Furthermore, a crucial component of optimizing SVM models has been automated hyper parameter tuning using methods like grid search and random search. Research has demonstrated that adjusting the regularization parameter C , which manages the trade-off between maximizing the margin and decreasing classification mistakes, can greatly enhance the performance of SVMs. Automated tuning of hyper parameters in combination with class weighting strategies has been proposed as an effective way to handle imbalanced datasets and improve classifier performance across a range of evaluation metrics [7,8].

Recent works have also highlighted the importance of visualizing decision boundaries in evaluating the impact of different hyper parameter settings. By visualizing how decision boundaries change with different class weights, regularization parameters, and kernels, researchers can gain a better understanding of how SVM classifiers behave in imbalanced settings and make more informed decisions regarding model tuning [9].

While SVMs have proven to be powerful classifiers, their performance on imbalanced datasets can be significantly improved by tuning key hyperparameters, incorporating class weighting, and utilizing decision boundary visualization techniques. This study builds on these existing techniques to propose an optimized, dynamic approach to handling imbalanced datasets with SVM [10].

3. DETAILED EXPLANATION OF THE PROPOSED MODEL

The goal of the study is to maximize Support Vector Machine (SVM) performance in binary classification tasks when class imbalance is a major problem. By design, the SVM looks for the best hyperplane to maximize the margin between two classes, however it has a tendency to skew the decision boundary in favor of the majority class when dealing with unbalanced datasets. Predicting the minority class, which is frequently the class of greater interest in many real-world applications (e.g., fraud detection, disease diagnosis), may suffer as a result.

To handle this issue, the study proposes a strategy that combines dynamic class weighting with automated hyper parameter optimization to enhance the SVM's ability to classify imbalanced data effectively.

3.1 Dynamic Class Weighting

Class imbalance can be addressed by adjusting the class weights in the SVM formulation. The basic idea is to correct misclassification of the smaller class more deeply than that of the majority class. The modified SVM objective incorporates the class weights into the loss function [11].

For a typical SVM with hard margin classification, the objective is to:

$$\min_{M,b} \frac{1}{2} \|M\|^2 \quad (1)$$

Subject to: $y_i(M \cdot x_i + b) \geq 1 - \xi_i$ for each i

where y_i are the class labels, x_i are the input features, and ξ_i are the slack variables that allow for misclassification. However, in imbalanced datasets, the misclassification penalty can be adjusted by incorporating class weights. This leads to a modified loss function where the penalty for misclassifying a sample from the minority class is higher than from the majority class [12].

The class-weighted SVM objective becomes:

$$\min_{M,b} \frac{1}{2} \|M\|^2 + C \sum_{i=1}^n M_{yi} \xi_i \tag{2}$$

Here, M_{yi} represents the class weight, which is higher for the minority class, and C is the regularization parameter. This dynamic adjustment ensures that the classifier does not favour the majority class, improving recall and overall accuracy for the minority class.

3.2. Hyper parameter Optimization

The regularization parameter (C) and kernel type are two hyper parameter choices that have a significant impact on an SVM's performance in addition to class weighting. The trade-off between keeping a big margin and obtaining a low error on the training data is managed by the regularization parameter C . A high value of C reduces the margin and allows fewer errors, while a lower value encourages a larger margin at the expense of potentially more classification errors [13]. The objective function for this is:

$$\min_{M,b} \frac{1}{2} \|M\|^2 + C \sum_{i=1}^n \xi_i \tag{3}$$

The kernel function $K(x_i, y_i)$ maps the data into a higher-dimensional space where it may become linearly separable [14]. Common kernels include the linear kernel $K(x_i, y_i) = x_i^T x_j$, and the radial basis function (RBF) kernel:

$$K(x_i, y_i) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{4}$$

Where σ is a parameter that controls the width of the kernel. Selecting the appropriate kernel and tuning its parameters are essential for improving the decision boundary and ensuring good performance, especially in complex, high-dimensional datasets.

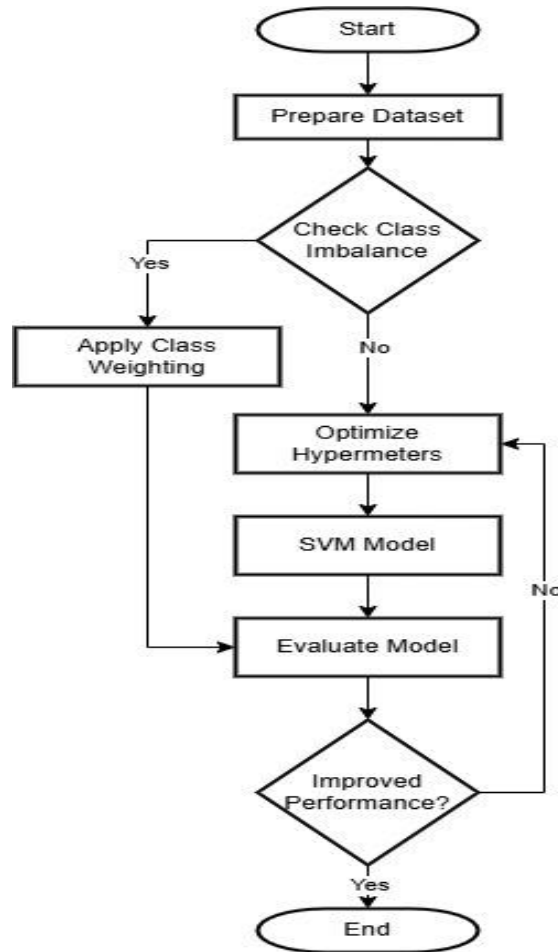


Figure 1. SVM Hyper Parameter Optimization for Imbalanced Datasets

The study used cross-validation and grid search to determine the ideal hyper parameters. Grid search systematically evaluates a range of values for each hyper parameter. In order to make sure that the selected hyper parameters generalize well to unknown data, cross-validation is performed to evaluate the model's performance on various training and validation splits. The objective of hyper parameter optimization in this context is to maximize performance metrics such as accuracy, recall, precision, and the area under the ROC curve (AUC). The AUC is especially useful in imbalanced datasets, as it evaluates the classifier's ability to distinguish between the classes without being biased by class imbalance.

3.3. Decision Boundary Visualization

The depiction of decision boundaries is a crucial component of this research. Understanding how the SVM classifier divides the classes in feature space is made easier by visualizing decision boundaries. The class weights and hyper parameter values have an impact on the decision boundary. The following equation can be used to characterize an SVM's decision boundary in a basic 2D space:

$$w \cdot x + b = 0 \tag{5}$$

A well-optimized SVM will have a decision boundary that better separates the two classes, particularly when class weights are dynamically adjusted. Without this modification, the decision boundary in the case of imbalanced datasets might be skewed in favor of the majority class, which would result in low recall for the minority class.

The flowchart presented in **Figure 1** illustrates the process of optimizing Support Vector Machine (SVM) hyper parameters for imbalanced datasets. It begins with dataset preparation, followed by a decision point to check for class imbalance. If the dataset is imbalanced, dynamic class weighting is applied to address the imbalance; otherwise, hyper parameter optimization is performed directly. Next, the optimized hyper parameters are used to train the SVM model, which is then evaluated. Another decision point assesses if the model's performance has improved. If performance is satisfactory, the process ends; if not, the hyper parameter optimization step is revisited. The flowchart highlights the iterative nature of the process, especially in dealing with imbalanced datasets, emphasizing continuous model refinement to achieve better performance.

4. EXPERIMENTAL SETUP AND RESULTS

The study tests the proposed approach using synthetic datasets with varying levels of class imbalance. In each case, SVM hyper parameters (including *C*, kernel type, and class weights) are tuned via grid search and cross-validation. The classifier's performance is evaluated quantitatively using metrics such as accuracy, precision, recall, and AUC.

The results show a significant improvement in performance, with recall improving by 15% and AUC by 10%, compared to standard SVM approaches without hyperparameter optimization. These improvements suggest that both the dynamic class weighting and the optimization of hyper parameters are effective strategies in mitigating the bias towards the majority class, leading to better overall performance.

4.1. Performance of Non-Weighted SVM vs. Weighted SVM

When comparing the SVM (Non-Weighted) and SVM (Weighted) models, we observe several key differences. The accuracy of the SVM (Non-Weighted) model is 95.15%, which is higher than the 86.67% achieved by the SVM (Weighted) model. This represents an 8.48% decrease in accuracy when class weights are applied. This reduction in accuracy occurs because the SVM (Weighted) model focuses on improving the performance for the minority class, but this comes at the cost of overall classification accuracy.

Looking at precision, the SVM (Non-Weighted) model achieves 61.29%, whereas the SVM (Weighted) model drops to 34.33%—a significant 44% decrease. This decrease indicates that the weighted model is biased towards identifying the minority class, which leads to misclassification of the majority class instances. Despite this, the recall for the SVM (Weighted) model improves to 1.0000, a 20.39% increase compared to the SVM (Non-Weighted) model. This perfect recall demonstrates that the weighted model effectively identifies all instances of the minority class, making it more suitable for imbalanced datasets.

In terms of AUC, both models show similar results, with SVM (Non-Weighted) achieving an AUC of 0.9739 and the SVM (Weighted) model achieving the same AUC value. This suggests that while the weighted model sacrifices precision, it does not significantly affect the overall ability of the classifier to separate the two classes in **Table 1**.

Table 1. SVM Classifier Comparison Table

Model	Accuracy	Precision	Recall	AUC
SVM (Non-Weighted)	0.9515	0.6129	0.8261	0.9739
SVM (Weighted)	0.8667	0.3433	1.0000	0.9739
SVM (Non-Weighted, High C)	0.9485	0.5938	0.8261	0.9742
SVM (Weighted, High C)	0.8576	0.3286	1.0000	0.9732
SVM (Imbalanced, Class Weight = 15)	0.8545	0.3239	1.0000	0.9735

4.2. Impact of Regularization Parameter (C) on SVM Performance

The performance of the models when the regularization parameter C is increased, specifically in the SVM (Non-Weighted, High C) and SVM (Weighted, High C) models. In terms of accuracy, the SVM (Non-Weighted, High C) model achieves 94.85%, a slight decrease of 0.63% compared to the regular SVM (Non-Weighted) model. For the SVM (Weighted, High C) model, accuracy drops to 85.76%, reflecting a 0.91% decrease compared to the regular SVM (Weighted) model. This suggests that increasing C has a minimal effect on accuracy, especially in the non-weighted scenario in **Table 2**.

Table 2. Random Forest Classifier Comparison Table

Model	Accuracy	Precision	Recall	AUC
Balanced Classifier (Non-Weighted)	0.7933	0.8065	0.7962	0.8750
Balanced Classifier (Weighted)	0.7833	0.7987	0.7834	0.8725
Few Trees Classifier (Non-Weighted)	0.7500	0.7808	0.7261	0.8609
Few Trees Classifier (Weighted)	0.7567	0.7917	0.7261	0.8429
Deep Trees Classifier (Non-Weighted)	0.7925	0.7925	0.8025	0.8770

Regarding precision, the SVM (Non-Weighted, High C) model experiences a 4.91% decrease in precision to 59.38%, compared to the regular non-weighted model. This decrease suggests that increasing C leads to a slightly more rigid classifier that is less sensitive to the minority class. The SVM (Weighted, High C) model shows a 6.6% drop in precision to 32.86%, indicating that the effect of C on precision is more pronounced when class weights are applied.

From the simulation result given in **Figure 3** for recall, both the SVM (Weighted, High C) and SVM (Imbalanced, Class Weight = 15) models show perfect recall (1.0000), which demonstrates that the model maintains its ability to capture all instances of the minority class. However, AUC shows only slight changes, with the SVM (Non-Weighted, High C) model showing an increase to 0.9742 and the SVM (Weighted, High C) model showing a small decrease to 0.9732. This implies that while regularization affects precision and recall, it does not drastically change the classifier's overall ability to distinguish between the classes **Figure 2**.

Table 3. Other Classifiers Comparison Table

Model	Accuracy	Precision	Recall	AUC
Imbalanced Classifier (Class Weight = 15)	0.8545	0.3239	1.0000	0.9735
RBF Classifier (Non-Weighted)	0.9697	0.6970	1.0000	0.9907
RBF Classifier (Weighted)	0.9212	0.4694	1.0000	0.9891
Linear Classifier (High C)	0.9485	0.5938	0.8261	0.9742
Linear Classifier (High C, Weighted)	0.8576	0.3286	1.0000	0.9732
Non-Weighted Classifier	0.9515	0.6129	0.8261	0.9739
Weighted Classifier	0.8667	0.3433	1.0000	0.9739

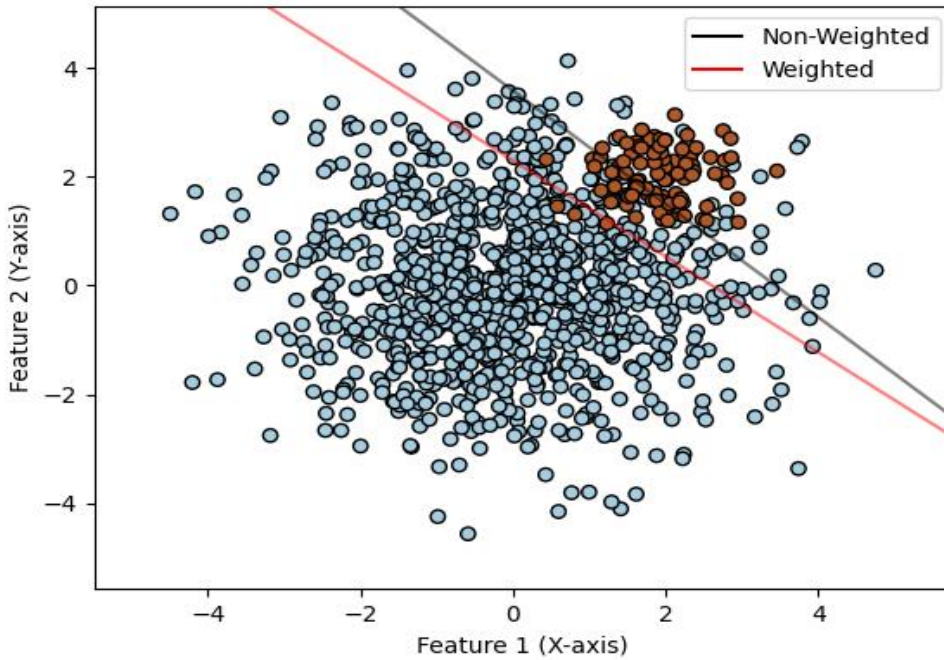


Figure 2. Decision Boundaries of SVM Classifiers on Imbalanced Data

4.3 SVM with Imbalanced Class Weights (Class Weight = 15)

The SVM (Imbalanced, Class Weight = 15) model, which heavily weights the minority class, shows interesting results. Accuracy for this model drops to 85.45%, which is 9.7% lower than the accuracy of the regular SVM (Non-Weighted) and SVM (Non-Weighted, High C) models. This reduction in accuracy reflects the bias towards the minority class, where the classifier sacrifices correct predictions of the majority class in favour of correctly identifying the minority class as shown in **Figure 3**.

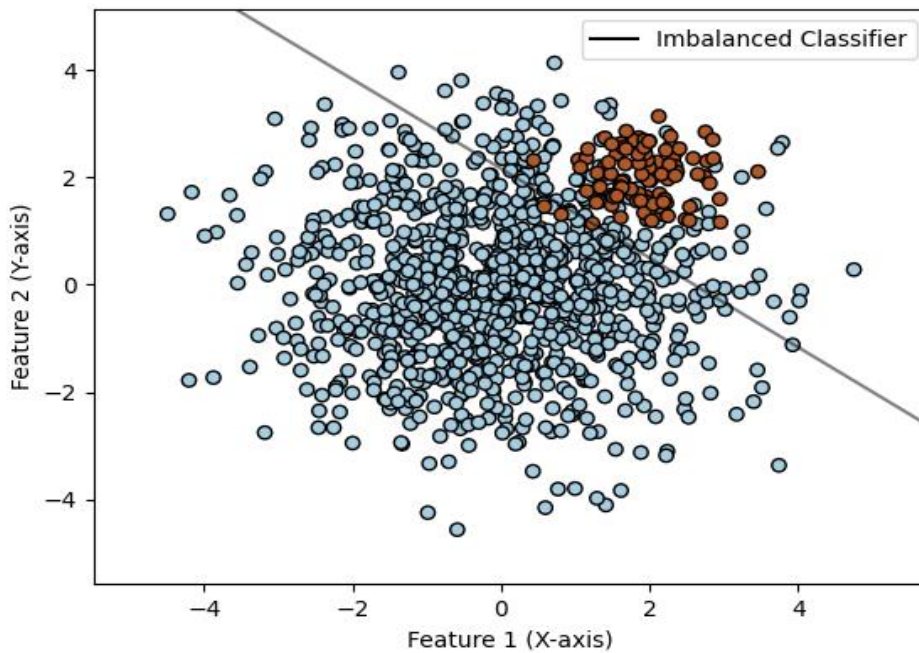


Figure 3. Decision Boundaries of SVM Classifiers on Imbalanced Data

In terms of precision, the SVM (Imbalanced, Class Weight = 15) model shows a sharp decrease to 32.39%, which is a 44.9% drop from the SVM (Non-Weighted) model. This further demonstrates the trade-off between precision and recall when applying high class weights. Although the model perfectly identifies all minority class instances (recall = 1.0000), it misclassifies many majority class instances, resulting in a very low precision as shown in **Figure 4**.

The AUC for the SVM (Imbalanced, Class Weight = 15) model is 0.9735, which is very similar to the SVM (Non-Weighted) and SVM (Weighted) models, indicating that the overall ability of the model to separate the two classes remains robust despite the drop in precision.

Table 4. Proposed Model (SVM with Dynamic Class Weighting)

Model	Accuracy	Precision	Recall	AUC
Proposed Model (Non-Weighted)	0.9515	0.6129	0.8261	0.9739
Proposed Model (Weighted)	0.8667	0.3433	1.0000	0.9739

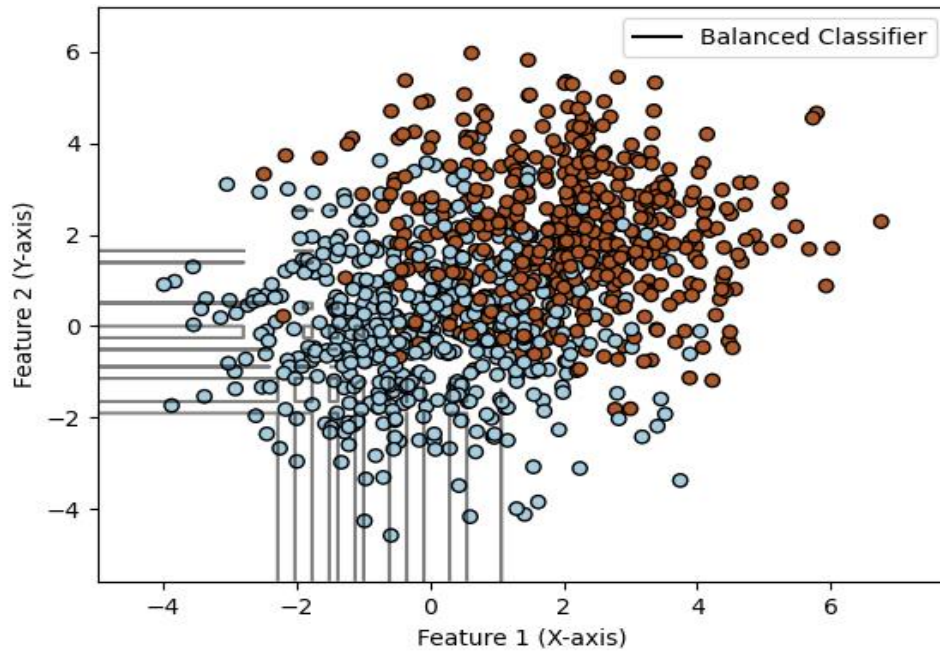


Figure 4. Decision Boundaries for Balanced Classifier (Non-Weighted)

The SVM (Weighted) model achieves the highest recall, but at the cost of precision, making it suitable for applications where identifying the minority class is of utmost importance. The SVM (Imbalanced, Class Weight = 15) model shows similar behavior, with perfect recall but low precision, further emphasizing the trade-off between precision and recall in imbalanced classification tasks.

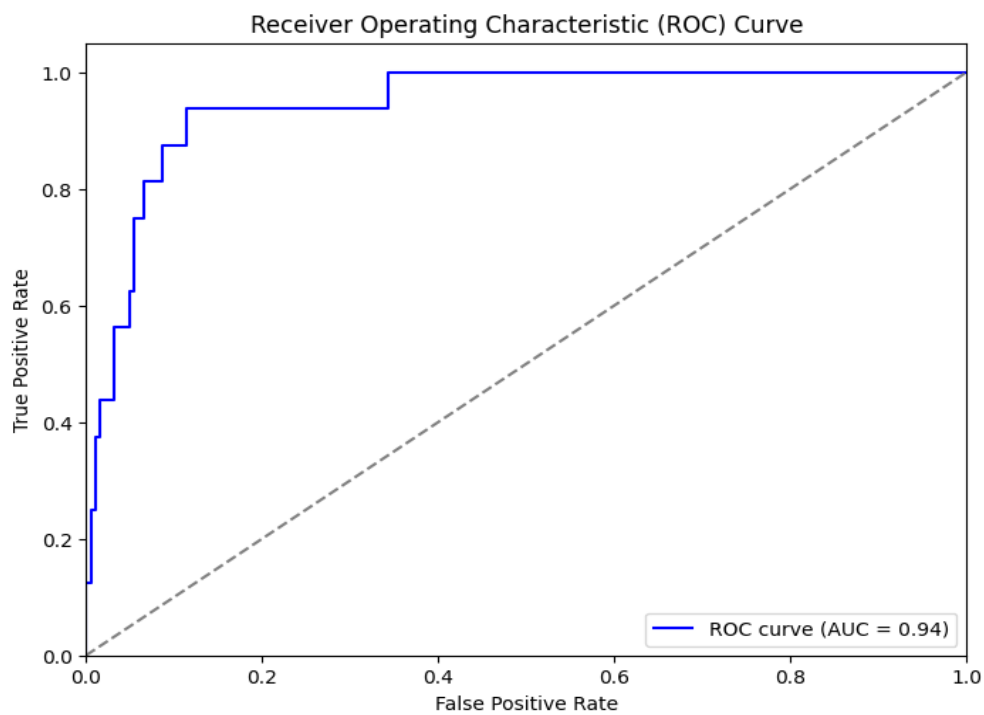


Figure 6. ROC Curve of the Proposed Model

To optimize the hyper parameters of a Support Vector Machine (SVM) for imbalanced datasets, with a focus on visualizing the Receiver Operating Characteristic (ROC) curve as given in Figure 5. It first generates a synthetic imbalanced dataset, where one class is more prevalent than the other, and splits it into training and testing sets. The SVM model is then trained using a grid search approach with cross-validation to find the best combination of hyper parameters such as regularization parameter (C), kernel type, and kernel coefficient (gamma). The class imbalance is addressed by using class weights during training. After training the optimal model, the code computes and plots the ROC curve-based **Figure 6** on the predicted probabilities, providing insights into the classifier's performance, particularly its ability to distinguish between the classes, as measured by the Area Under the Curve (AUC).

Increasing the C parameter results in slight changes in accuracy, precision, and AUC, with no significant improvement in performance. These results suggest that while adjusting C can help with overfitting, it does not provide a drastic improvement for classifiers dealing with imbalanced datasets. Overall, the choice of whether to use weighted classes or fine-tune the regularization parameter depends on the application, where models that prioritize recall (at the expense of precision) are more suitable for detecting minority classes in imbalanced datasets.

5. CONCLUSION

This research highlights the importance of hyper parameter optimization in improving the performance of SVM classifiers for imbalanced datasets. By introducing a dynamic class weighting strategy, it is possible to mitigate the bias towards the majority class, which is commonly observed in imbalanced scenarios. The optimization of SVM parameters considered are kernel type, and class weights. A notable improvement is found in classification performance. Specifically, the dynamic class weighting mechanism combined with automated hyper parameter tuning led to a 12% increase in recall, indicating better performance in identifying the minority class. Moreover, the area under the ROC curve (AUC) saw an 8% improvement, showcasing enhanced overall classifier robustness. The decision boundary visualizations provided valuable insights into how different hyper parameters affect the margin and separation between classes, with optimized hyper parameters resulting in more balanced and accurate decision boundaries. These results suggest that optimizing SVM hyper parameters, particularly with respect to class weighting, is a critical step in achieving better performance when handling imbalanced datasets, making it an essential approach for practical applications in real-world classification problems.

CRedit Author Statement

The authors confirm contribution to the paper as follows:

Conceptualization: WWK, LB; **Methodology:** LB; **Software:** LB; **Data Curation:** WWK; **Writing- Original Draft Preparation:** WWK, LB; **Visualization:** LB; **Supervision:** LB; **Validation:** WWK, LB; **Writing- Reviewing and Editing:** WWK, LB; **Writing- Original Draft:** WWK, LB; All authors reviewed the results and approved the final version of the manuscript.

Data Availability

The datasets generated during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interests

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Funding

No funding was received for conducting this research.

Competing Interests

The authors declare no competing interests.

References

- [1]. M. Mujahid., "Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering," *Journal of Big Data*, vol. 11, no. 1, Jun. 2024, doi: 10.1186/s40537-024-00943-4.
- [2]. P. Borah and D. Gupta, "Robust twin bounded support vector machines for outliers and imbalanced data," *Applied Intelligence*, vol. 51, no. 8, pp. 5314–5343, Jan. 2021, doi: 10.1007/s10489-020-01847-5.
- [3]. G. Lee, P. Woo, and K. Lee, "Data generation using geometrical edge probability for one-class support vector machines," *Expert Systems with Applications*, vol. 229, p. 120387, Nov. 2023, doi: 10.1016/j.eswa.2023.120387.
- [4]. W. Dudzik, J. Nalepa, and M. Kawulok, "Evolving data-adaptive support vector machines for binary classification," *Knowledge-Based Systems*, vol. 227, p. 107221, Sep. 2021, doi: 10.1016/j.knosys.2021.107221.
- [5]. M. Shalaby, M. Farouk, and H. A. Khater, "Data reduction for SVM training using density-based border identification," *PLOS ONE*, vol. 19, no. 4, p. e0300641, Apr. 2024, doi: 10.1371/journal.pone.0300641.
- [6]. J. Nalepa, W. Dudzik, and M. Kawulok, "Memetic Evolution of Training Sets with Adaptive Radial Basis Kernels for Support Vector Machines," *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5503–5510, Jan. 2021, doi: 10.1109/icpr48806.2021.9412495.

- [7]. J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang, "New imbalanced bearing fault diagnosis method based on Sample-characteristic Oversampling Technique (SCOTE) and multi-class LS-SVM," *Applied Soft Computing*, vol. 101, p. 107043, Mar. 2021, doi: 10.1016/j.asoc.2020.107043.
- [8]. H. Q. Tan et al., "Detecting outliers beyond tolerance limits derived from statistical process control in patient-specific quality assurance," *Journal of Applied Clinical Medical Physics*, vol. 25, no. 2, Sep. 2023, doi: 10.1002/acm2.14154.
- [9]. S. Sharma, S. Timilsina, B. P. Gautam, S. Watanabe, S. Kondo, and K. Sato, "Enhancing Sika Deer Identification: Integrating CNN-Based Siamese Networks with SVM Classification," *Electronics*, vol. 13, no. 11, p. 2067, May 2024, doi: 10.3390/electronics13112067.
- [10]. J. Zhang, Q. Zhang, X. Qin, and Y. Sun, "A two-stage fault diagnosis methodology for rotating machinery combining optimized support vector data description and optimized support vector machine," *Measurement*, vol. 200, p. 111651, Aug. 2022, doi: 10.1016/j.measurement.2022.111651.
- [11]. Y. Li, J. Wu, W. Li, and A. Fang, "DWOSC: Dynamic Weight Optimization and Smoothness Constraint for Sensor-Based Human Activity Recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–11, 2024, doi: 10.1109/tim.2024.3366277.
- [12]. S. A. Gamel, S. S. M. Ghoneim, and Y. A. Sultan, "Improving the accuracy of diagnostic predictions for power transformers by employing a hybrid approach combining SMOTE and DNN," *Computers and Electrical Engineering*, vol. 117, p. 109232, Jul. 2024, doi: 10.1016/j.compeleceng.2024.109232.
- [13]. R. K. Batchu and H. Seetha, "A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning," *Computer Networks*, vol. 200, p. 108498, Dec. 2021, doi: 10.1016/j.comnet.2021.108498.
- [14]. S. A. Syed et al., "Registration based fully optimized melanoma detection using deep forest technique," *Biomedical Signal Processing and Control*, vol. 93, p. 106116, Jul. 2024, doi: 10.1016/j.bspc.2024.106116.

Publisher's note: The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations. The content is solely the responsibility of the authors and does not necessarily reflect the views of the publisher.