Synopsis of Thesis on

Optimizing Fast Randomized One-Class Classification (FROCC) for Enhanced Binary Classification

Department Of Computer Science And Engineering

INDIAN INSTITUTE OF TECHNOLOGY DELHI

July 2024

Abstract

The Fast Randomized One-Class Classification (FROCC) model is a novel and efficient algorithm for binary classification tasks, known for its simplicity and effectiveness. This thesis explores various enhancements and optimizations to the FROCC model, aiming to improve its performance and robustness. The research is structured into several key phases, each addressing different aspects of model optimization and comparison with traditional baseline models.

Initially, the study investigates the impact of tuning key hyperparameters, such as dimension and epsilon, on the performance of the FROCC model. Through extensive experimentation on multiple binary datasets, optimal configurations are identified, leading to significant performance gains. A combined decision function is then proposed, integrating the strengths of two independently trained FROCC models to enhance classification accuracy.

To further refine the model, vector optimization techniques are employed. By ranking and prioritizing the most effective classifier dimensions, the decision function is optimized to use only the top-performing vectors, resulting in improved predictive accuracy. The optimized FROCC model is rigorously evaluated against traditional baseline models, including Support Vector Machine (SVM) and CatBoost, demonstrating competitive or superior performance.

This thesis also explores various kernel functions to enhance the FROCC model further. The results underscore the potential of FROCC as a robust and efficient alternative to traditional binary classifiers, with significant improvements achievable through targeted optimizations.

The findings of this research contribute to the field of machine learning by providing a comprehensive understanding of FROCC's capabilities and limitations, along with practical strategies for its enhancement. The proposed methods and insights pave the way for future research and applications of FROCC in diverse binary classification tasks.

KEYWORDS: Machine learning, binary classification, FROCC

Contents

1 Introduction **3**

1.1 Background and Motivation 3

1.2 Objectives 3

1.3 Scope of the Study 4

2 Review of Binary Classification Methods **5**

2.1 Binary Classification and Traditional Methods 5

2.2 One-Class Classification and FROCC 6

2.3 Combination Models 6

2.4 Hyperparameter Optimization 7

2.5 Evaluation Metrics 8

2.6 Baseline Models: SVM and CatBoost 9

3 Methodology **10**

3.1 Introduction 10

3.2 Dataset Description 10

3.3 Dataset Preprocessing 10

3.4 Model Training 11

3.5 Combining Model Outputs 11

3.6 Vector Optimization 11

3.7 Baseline Models 12

3.8 Experimental Setup 12

4 Experimental Results **14**

5 Conclusion and Future Works **17**

5.1 Discussion 17

5.2 Limitations 17

5.3 Future Work 18

5.4 Conclusion 18

1 Introduction

1.1 Background and Motivation

Binary classification is one of the four main classification tasks in machine learning. In a binary classification task, the goal is to classify the input data into two different categories. In this case, we use labels such as true and false, positive and negative, etc. For example, we might want to detect whether a given image is of a dog or a cat.

FROCC (Fast Random projection-based One-Class Classification) [BVBB21] is an algorithm for one-class classification that provides a powerful classifier by applying a simplified envelope method to random projections of training data. Here are its key features and advantages:

- This algorithm is very fast so, suitable for large datasets. Also, this makes it suitable for real-time applications.
- It is robust to noise and outliers.
- It provides results that are very easy to understand and suitable for various operations for comparisons.
- It works well with large datasets.
- It works well with different types of data

So, FROCC is a powerful tool that balances speed, robustness, and scalability.

The key challenges in the case of binary classification are imbalanced data and noisy data but FROCC can handle these to some extent. So, we want to take these advantages and improve on them.

1.2 Objectives

- **Primary Objectives :**
  - To develop and evaluate the combination of FROCC models for better performance.

- Compare the combined models with baseline models such as SVM and CatBoost.

- **Secondary Objectives :**

  - Investigate the impact of various kernel functions on the performance of the models.
  - Optimise the hyper-parameters for FROCC to improve accuracy.
  - Explore vector optimization to improve model predictions.

1.3 Scope of the Study

1. **Datasets :** The datasets used in the experiments are OpenML Datasets. OpenML datasets are uniformly formatted and come with rich meta-data to allow automated processing. We selected several binary datasets with varying numbers of features and training samples for the experiments. The datasets used in the experiments with their descriptions are mentioned in Table 1.

2. **Methodology :**

   - We train the FROCC models on different datasets while applying a few optimization techniques for enhancing our combination models.
   - We compare the accuracy results of the individual models and combined models with the baseline models.
   - Our evaluation of the results on several datasets shows that our combined models outperform the individual FROCC models.

3. **Structure of Thesis :** We discuss the related literature in Sec. 2. Our method and techniques are in Sec. 3. Experimental setups and results are in Sec. 4. Conclusion is in Sec. 5.

Dataset Id	Name	Features	Training Samples
15	breast-w	9	699
29	credit-approval	15	690
31	credit-g	20	1000
37	diabetes	8	768
50	tic-tac-toe	9	958
1063	kc2	21	522
1464	blood-transfusion-service-center	4	748
1480	ilpd	10	583
1510	wdbc	30	569
6332	cylinder-bands	37	540
23383	dresses-sales	12	500
40994	climate-model-simulation-crashes	18	540

Table 1: OpenML Datasets

2 Review of Binary Classification Methods

2.1 Binary Classification and Traditional Methods

Binary classification is a problem where the outcomes are restricted to two possible outcomes. It is widely used in email spam detection, credit card fraud detection, and medical diagnosis such as detecting cancer cells [LK24].

- **Support Vector Machines (SVM) :** It is a supervised machine learning algorithm. It finds the optimal hyperplane that can separate the data points into different classes. The dimension of the hyperplane depends on the number of features. It is very efficient in the case of high-dimensional cases. It is also memory efficient as it only uses subsets of the training points in the decision function. It cannot handle missing values in the dataset. The SVM algorithm is not suitable for large data sets, overlapping data, and when the number of features exceeds the number of training samples.

- **CatBoost :** It is a gradient boosting technique and provides high accuracy in classification problems. It constructs decision trees on each iteration. It uses regularization to avoid overfitting. The main limitation of CatBoost is memory consumption. It requires significant memory resources in case of large datasets or high dimensional data.

2.2 One-Class Classification and FROCC

One-class classification (OCC) involves training the model on normal data, and then predicting whether the new data is normal or an anomaly. It can be used to monitor motor failure and the operational status of nuclear power plants as normal.

- **Fast Randomized One-Class Classification:** It is a scalable and efficient method for one-class classification that uses random projections and interval-based decision boundaries. FROCC provides a very good multiple of speed up in terms of training and testing as compared to other traditional models.

2.3 Combination Models

Combining models, also referred to as ensemble methods, is a powerful technique to achieve better performance than the individual models. There are many approaches in Ensemble models to perform a particular task. There are several strategies for combining models depending on the need.

1. **Bagging**
    - Train multiple instances of the same model on different subsets of training data
    - Models are trained in parallel
    - Predictions from each model are combined
    - Reduces variance leading to stable and accurate predictions.
2. **Boosting**
    - Builds models sequentially
    - Each model attempts to correct errors of previous models
    - More weight is given to incorrect instances so more focus on difficult cases
    - Reduces bias and variance
3. **Stacking**

- Training base models and then train another model (meta-learner) to combine their predictions
- Base models can be of different types
- It is flexible and can incorporate different base models

4. **Voting Classifier**

    - Aggregates the predictions of multiple models to make a final prediction
    - Base models make independent predictions
    - The class getting the most votes gets chosen as the final prediction.

Combined models often lead to improved accuracy, robustness, and generalization. Combining models can be computationally expensive and time-consuming. The results of combining models may be harder to interpret.

2.4 Hyperparameter Optimization

In the machine learning pipeline, hyperparameter optimization is a crucial step. This impacts model performance and generalization. These parameters are set before the training process. Effective hyperparameter optimization can enhance models' predictive power and robustness. There are several techniques for hyperparameter optimization.

1. **Grid Search**

    - Exhaustive search over a predefined set of hyperparameters
    - Ensures all combinations are considered
    - Very time-consuming and resource-intensive

2. **Random Search**

    - Selects values randomly from a defined distribution
    - Only evaluates a fixed number of combination
    - Fast and practical for high-dimensional data
    - May miss the optimal combination since it does not evaluate all possible combinations

3. **Bayesian Optimization**

    - probabilistic model of the objective function and uses it to select the most promising hyperparameters to evaluate
    - requires fewer evaluations to find optimal hyperparameters
    - Uses past evaluations to make informed decisions about the next set of hyperparameters
    - More complex as compared to the grid and the random search

4. **Genetic Algorithms**

    - Effective at exploring a large and complex hyperparameter space
    - Can adapt to different types of optimization problems
    - slow and requires significant computational resources

2.5 Evaluation Metrics

Evaluation metrics are essential tools for assessing the performance of machine learning models. Different tasks and models need different evaluation metrics to measure their effectiveness. We discuss the evaluation metrics used in this work.

1. **Accuracy:** The ratio of correctly predicted instances to the total instances. Suitable for balanced datasets.

2. **Precision:** The ratio of true positive predictions to the total positive predictions. It is important where the cost of false positives is high.

3. **Recall:** The ratio of true positive predictions to the total actual positives. It is important where missing a true positive is high.

4. **ROC-AUC (Receiver Operating Characteristic - Area Under Curve):** Measures the ability of the model to distinguish between classes. This curve plots the positive rate against the negative rate.

2.6 Baseline Models: SVM and CatBoost

Baseline models provide benchmarks for comparing the models. Two powerful and widely used models are Support Vector Machines (SVM) and CatBoost. Each has its features and advantages.

1. **Support Vector Machines**
    - Effective in high-dimensional spaces and cases where the number of features exceeds the sample space
    - Aims to find the optimal hyperplane that separates the classes of the feature space.
    - Uses kernel functions to transform the input data into higher-dimensional space.
    - Takes a regularization parameter that controls the trade-off between maximizing the margin and maximizing the classification errors.
    - Can be computationally intensive with large datasets.
    - Choosing the right kernel and regularization parameters is crucial for optimal performance.
2. **CatBoost**
    - Designed to handle categorical data effectively
    - Requires minimum parameter tuning
    - Optimized for speed and efficiency
    - Have built-in mechanisms to control overfitting
    - Can be memory-intensive in case of large datasets
    - Training can be time-consuming compared to simple models
3. **Compare Baseline models**
    - CatBoost excels with datasets that have categorical features while SVM is suited for more numerical and high-dimensional data.
    - CatBoost is more user-friendly as it requires less manual preprocessing and parameter tuning.
    - CatBoost often provides better results due to its categorical data

3 Methodology

3.1 Introduction

This chapter details the methodologies employed in the study, focusing on the design, implementation, and optimization of FROCC models for binary classification. It also explains the process of combining the outputs of two FROCC models and the techniques used for evaluating and comparing their performance against baseline models like SVM and CatBoost.

3.2 Dataset Description

The study utilizes multiple binary classification datasets identified by their OpenMl dataset IDs: 23381, 1063, 40994, 6332, 1510, 1480, 29, 15, 1464, 37, 50, 31. The brief details of the dataset are provided in the table 1

3.3 Dataset Preprocessing

The dataset is downloaded using the API provided by OpenML. Each dataset has different types of features. We use the One-Hot Encoder to encode the non-numerical features to numerical values. We split the training data into two parts depending on the class. The labels are also converted to numerical class (0 or 1)

- The dataset is downloaded using the API provided by OpenML.
- Use the One-Hot Encoder to encode the non-numerical features to numerical values
- Split the data into training and test data
- Encode labels into positive class and negative class
- Split the training data for positive class and negative class

3.4 Model Training

- The Positive class model is trained on positive data points
- The Negative class model is trained on negative data points
- Using the grid search and random search the parameters for dimensions and interval width (epsilon) are optimized.
- Model performance is evaluated based on ROC AUC score.

3.5 Combining Model Outputs

- Define a combined function to use the results from the positive and negative class models.
- We use the weights to combine the outputs into a combination function.

$$CombinedScore = (w1 * positveScore + w2 * negativeScore)/(w1 + w2)$$

- For effective combination the inverse of the score of the negative class model is considered
- We use Sequential Least Squares Programming (SLSQP) and Binary Cross-Entropy Loss to find optimal weights.
- Selection of initial weights impacts the finding of optimal weights so use several combinations of initial weights.
- Comparison of SLSQP and Binary Cross-Entropy Loss is in Figure

3.6 Vector Optimization

In the context of the FROCC model, vectors refer to the projections of data points. These vectors are crucial for the decision function. Optimizing these vectors significantly improves the model's performance.

- Each data point is projected onto the multiple vectors defined by the model

- For each projection, check if the projected value falls within the predefined intervals
- Score will be produced by calculating the fraction of projections that agree with the intervals
- We rank the vectors according to the frequency of their correct predictions
- Only consider the vectors above a threshold for the decision function

3.7 Baseline Models

For evaluating the performance of the combined FROCC models, it is essential to compare it with baseline models. In this section, we train baseline models SVM and CatBoost.

1. **Support Vector Machine (SVM)**
    - We use the library scikit-learn for the SVM model
    - We use the complete training data with both classes for training the model
    - We record the ROC AUC score for comparison with the combined model
2. **CatBoost**
    - We use the library CatBoost for the model
    - We use the complete training data with both classes for training the model
    - We record the ROC AUC score for comparison with the combined model

3.8 Experimental Setup

The experimental setup is a crucial aspect of this study for systematic evaluation and comparison of combined FROCC models against the baseline models.

- **Software and Tools**
    - **Python:** We used Python as the primary programming language for implementing the models.
    - **scikit-learn:** Used for implementing baseline models (SVM) and for preprocessing steps such as normalization and encoding

  - **PyTorch:** Used for optimization tasks.
  - **CatBoost:** Used for implementing the CatBoost model.
  - **NumPy:** Employed for numerical operations and handling arrays that are used for data and model manipulation.
  - **Pandas:** Used for data manipulation and analysis.
  - **Matplotlib:** Used for plotting and visualizing the results.

- **Hardware:** The hardware used for this experiment is
  - **Chip:** Apple M3
  - **Memory:** 16 GB
  - **Cores:** 8-core CPU and 8-core GPU

4 Experimental Results

In this section, we present the experimental results obtained from evaluating individual FROCC models, combined FROCC models, and the baseline models. We also discuss the impact of different optimization techniques.

1. **Impact of Different Kernels:** Different kernels were used to assess the performance of the FROCC models. The kernels used include Linear, Polynomial, Radial basis function (RBF), and Sigmoid kernels. The experiment reveals RBF kernel worked better for FROCC models.

2. **Dimension and Epsilon Optimization:** We used grid search and Bayesian optimization techniques. These parameters varied across datasets highlighting the need for dataset-specific tuning.

3. **Weights Optimization:** We used techniques such as SLSQP 4 and Binary cross entropy optimization 3. The results show that SLSQP is more suitable for combined FROCC models.

4. **Vector Optimization:** Identified the most influential vectors in the decision functions. This improved the performance of the models. Comparison results of FROCC models and baseline models are shown in 1 and 2

5. **Comparison with Baseline Models:** FROCC models were compared baseline models to benchmark their performance. While the performance of the combined model showed promising results, it did not always perform better than the baseline models. This indicates we need to do further refinement for better results.

6. **Robustness:** Multiple initial weights were tested to ensure the robustness of FROCC models. The results show careful selection of initial weights is essential for optimal performance.

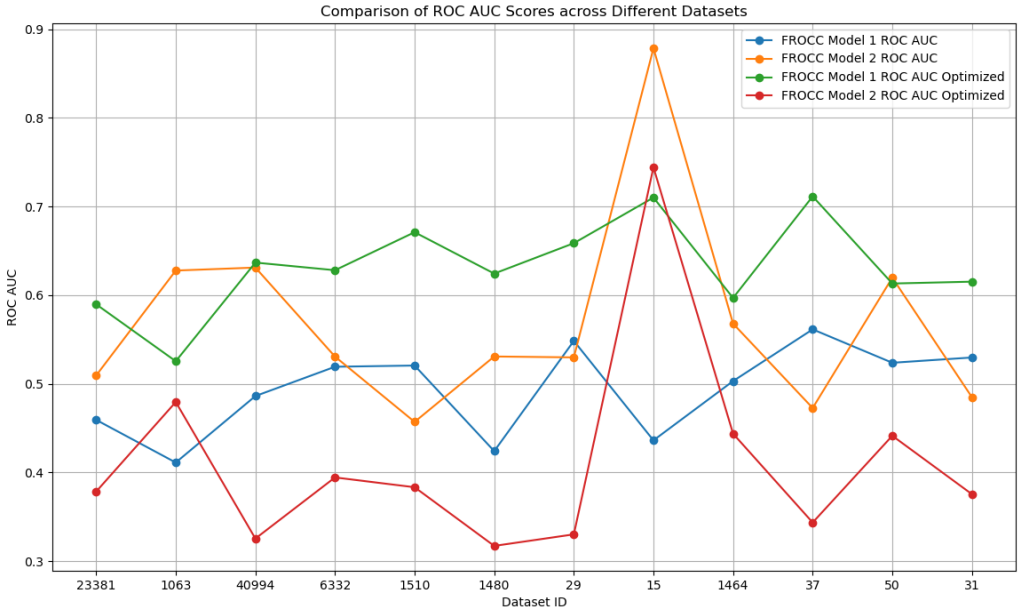

Figure 1: Comparison after Vector Optimization

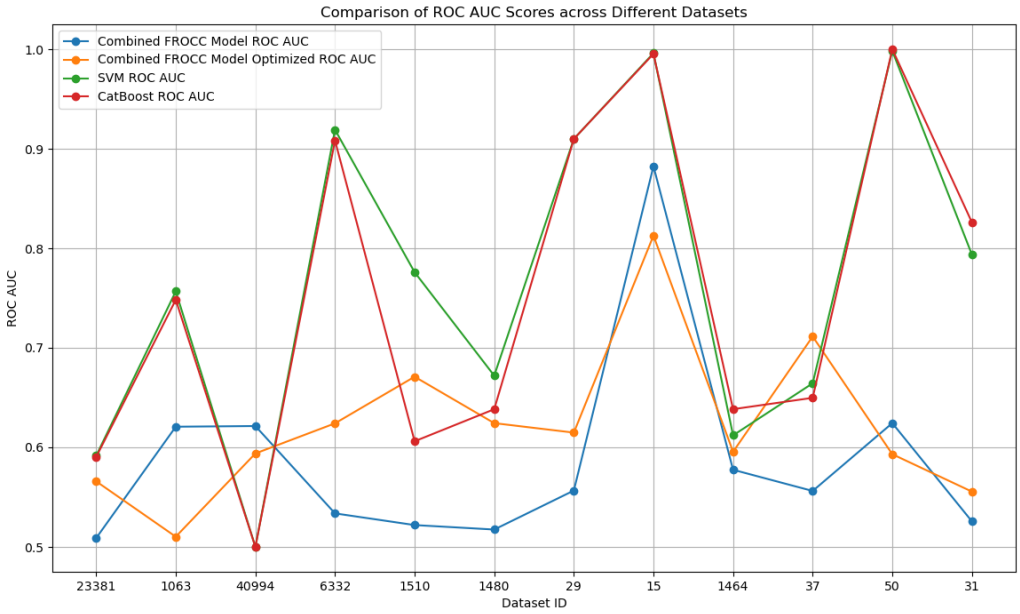

Figure 2: Comparison after Vector Optimization

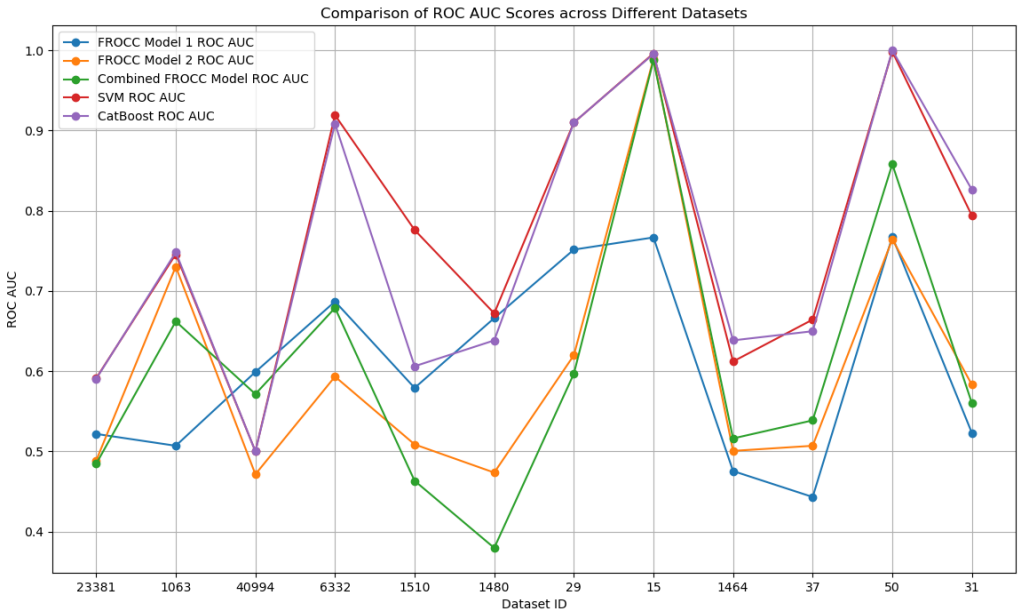

Figure 3: Weight optimization using binary entropy cross

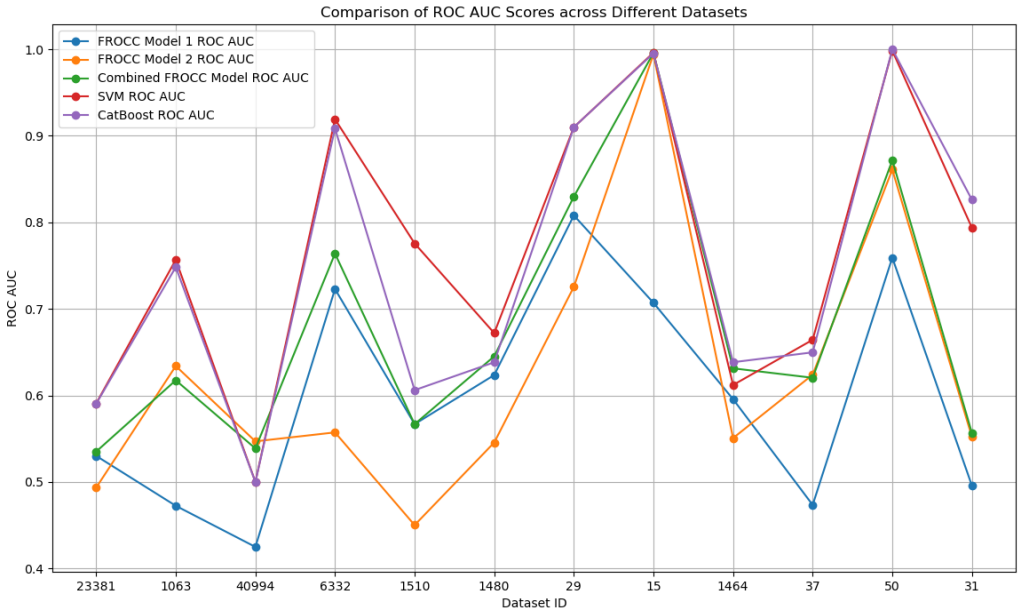

Figure 4: Weight optimization using SLSQP

5 Conclusion and Future Works

In this section, we discuss the implications of our findings and propose future work to further improve the FROCC model's performance.

5.1 Discussion

1. **Performance of Combined Models:** The combined model outperforms the individual models. This shows that combining models can leverage the strengths of both models for more accurate predictions.

2. **Kernel Function:** The choice of kernel function affects the performance of the model. The RBF kernel provides better results, so it is more suitable for FROCC models. This also shows that the choice of a suitable kernel function depending on dataset characteristics is necessary.

3. **Hyperparameter Tuning:** Choice of dimension and epsilon parameter was crucial for the performance of the FROCC model. Techniques like grid search and Bayesian optimization were very effective in our case.

4. **Vector Optimization:** Identifying and using the most influential vectors can improve the model performance.

5. **Comparison with Baseline Models:** While the combined model outperformed the individual models it did not outperform baseline models in most cases. It shows the potential for the FROCC model but further refinement and optimization are necessary.

5.2 Limitations

Several limitations were encountered during the experiment:

1. **Computation Complexity:** Hyperparameter tuning for the FROCC model is computationally intensive, indicating not efficient in the case of larger datasets.

2. **Parameter Sensitivity:** Performance of the FROCC model depends on the initial choice of parameters such as dimension, epsilon, and initial weights. This requires intensive parameter tuning.

3. **Generalization:** We tested the combination model on limited datasets in this study. Further experiments are needed to assess its applicability to different datasets.

5.3 Future Work

We propose the following directions for future work:

1. **Advanced Hyperparameter Optimization:** We need to implement more advanced techniques such as automated machine learning for an efficient tuning process.

2. **Scalability Enhancement:** Need to find ways to reduce the computational complexity such as parallel processing, efficient data sampling methods, etc. such that we can handle larger datasets.

3. **Ensemble Methods:** Need to try more ensemble methods for better leveraging the results of the FROCC models.

4. **Transparency:** We need better ways to visualize and understand the decision-making process. This will help its adoption to real-world applications.

5.4 Conclusion

Our study shows the potential of the FROCC model and its combination for binary classification. By effective optimization of parameters, FROCC can compete with traditional models. However, further refinement and optimization are required to outperform traditional and widely used models. Future work should focus on solving the limitations mentioned.

REFERENCES

[BVBB21] Arindam Bhattacharya, Sumanth Varambally, Amitabha Bagchi, and Srikanta Bedathur. Frocc: Fast random projection-based one-class classification, 2021.

[LK24] Chongmin Lee and Jihie Kim. Biomarker based cancer classification using an ensemble with pre-trained models, 2024.