# Comprehensive study of Hyperparameter optimisation models

**KRISHNA CHAITHANYA MARRIPATI[1], MERY HARIKA GADDAM[1], AND SAI VIKAS REDDY YEDDULAMALA[1].**
[1]Computer Science Department, North Carolina State University, Raleigh, NC 27695, USA (e-mails: kmarrip@ncsu.edu, mgaddam@ncsu.edu, syeddul@ncsu.edu)

**ABSTRACT** This project focuses on improving Support Vector Machine (SVM) regression models by fine-tuning two important parameters: the penalty parameter (C) and the slack variable for the loss function ($\epsilon$). We use three methods to optimize these parameters: grid search, Optuna, and Sequential Model Optimization (SMO). First, we use grid search, which involves trying out every possible combination of C and $\epsilon$. This gives us a baseline understanding of how well the model performs in different settings. Next, we use Optuna, a more efficient method. Optuna adjusts the search space dynamically based on the results we get, helping us find the best settings more quickly. Finally, we employ Sequential Model Optimization (SMO), a model-based approach. SMO uses only a small portion of the data. This reduces the computational workload significantly. By comparing these methods, we make the process of tuning the model's parameters smoother and improve its overall performance.

**INDEX TERMS**
Sequential Model Optimization (SMO), Optuna, Hyperparameter tuning, Support Vector Machine.

## I. INTRODUCTION

The primary issue addressed in this research is hyperparameter optimization in machine learning models. Hyperparameters are settings or configurations that are external to the model and must be set before training begins. These parameters significantly influence model behavior and performance. The core problem is determining the best set of hyperparameters, as they often have a non-linear relationship with model performance and are not directly learnable from the training data.

This topic is interesting because effectively tuned hyperparameters can drastically improve a model's performance. The importance lies in its wide-ranging applications, including areas such as healthcare for predictive diagnostics, finance for risk assessments, and automation in vehicles and robotics. In the context of SVMs, proper tuning of C and $\epsilon$ can balance the trade-off between achieving low training error and maintaining the model's ability to generalize on new data. Better hyperparameter settings lead to more accurate predictions, efficient resource use, and robust models that generalize well on unseen data.

The difficulty in hyperparameter optimization comes from the vastness and complexity of the hyperparameter space. Naive approaches like grid search attempt to explore this space exhaustively by evaluating a grid of parameter combinations, but they become impractical as the number of hyperparameters grows due to the 'curse of dimensionality'. These methods consume excessive computational resources and time, making them inefficient and often infeasible for large datasets or complex models.

Previous solutions, including grid search and random search, lack efficiency and often do not scale well with the dimensionality of the problem. Advanced techniques like Bayesian optimization improve over these by using probabilistic models to guide the search; however, they can be complex to implement and may require tuning of their own hyperparameters. Our approach uses Sequential Model Optimization (SMO) which dynamically adjusts the search based on performance feedback from trials. This method is more adaptive and scales better with problem complexity.

The key components of our approach include a model-based evaluation system that uses initial trial results to predict the most promising hyperparameters. This system significantly reduces the number of trials needed by focusing on more promising areas of the hyperparameter space. However, the approach assumes that initial trials are representative of larger trends, which may not always hold, especially in highly irregular parameter spaces.

As we progressed in our research, we encountered several key questions. These questions helped guide our study and understanding of how to optimize hyperparameters more

effectively in machine learning models:

**Research Question 1:** What is the impact of initial hyper-parameter selection on the effectiveness of the SVM model?
The impact of initial parameter selection on the effectiveness of an SVM model is significant. When parameters are chosen wisely at the start, they can lead to faster and more accurate convergence in the model's performance. Better hyperparameters often find the global optimum much fastaer. On the other hand, poor initial hyper-parameter choices might result in longer training times and potentially less optimal model performance. This highlights the importance of a strategic approach to selecting initial hyper-parameters in SVM models.

**Research Question 2:** How does the performance of grid search compare to more advanced techniques like Optuna and Sequential Model Optimization (SMO) in terms of computational efficiency and accuracy?
Grid search, while thorough and systematically explores all possible combinations of hyper-parameters, makes it computationally expensive and often impractical for large datasets or hyper-parameter spaces. Optuna and SMO, in contrast, use more sophisticated strategies to search the hyper-parameter space. Optuna utilizes Bayesian optimization techniques to selectively explore the space, thereby improving efficiency by focusing on more promising areas based on previous results. SMO takes this further by using a predictive model to estimate the performance of hyper-parameters using a subset of data, greatly reducing the computational burden. Empirical results from the project indicate that Optuna and SMO not only reduce the time required for hyper-parameter tuning compared to grid search but also potentially improve the accuracy by avoiding overfitting to extensive hyper-parameter combinations that grid search might entail.[3]

**Research Question 3:** What are the limitations of using a model like SMO for SVM regression (hyperparameter tuning), and how do these limitations affect the optimization outcomes?
The predictive model in SMO assumes that the subset of data used to train the model is representative of the entire dataset, which may not always be true. If the subset does not capture all the variability of the key characteristics of the data, the model's predictions could be biased or inaccurate. This misrepresentation can lead to suboptimal hyperparameter choices. Moreover, the model's effectiveness can be constrained by the choice of features, the type of learning algorithm used, and its parameter settings. These factors need careful consideration to ensure that the SMO approach effectively balances computational efficiency with the accuracy of hyperparameter optimization[7].

The contributions of this paper are: We introduced a novel Sequential Model Optimization technique for efficient hyper-parameter tuning. We provide empirical evidence through extensive experiments demonstrating that our method reduces computational costs and improves optimization efficacy compared to traditional methods. The research explores the scalability of our approach and discusses its practical implications for deploying advanced machine-learning models in real-world applications.

The rest of this paper is structured as follows: Section 2 consists of the related work plus a general overview of the field. Section 3 includes the algorithms used to compare the results, experimental setup, and different datasets used. Section 4 consists of the results, future scope, and conclusion based on the experimental results obtained.

## II. BACKGROUND

This section of the paper offers an overview of the work done so far in the field of machine learning, specifically focusing on the challenges and advancements in hyperparameter optimization. Before we dive into the details of our proposed Sequential Model Optimization (SMO) approach, it's important to understand the broader context of this research area.

Hyperparameter optimization is a crucial part of developing machine learning models. Hyperparameters are the settings and configurations that govern how a model learns from data. Unlike model parameters, hyperparameters are not learned during the training process but must be set before training. Correctly setting these hyperparameters can dramatically improve a model's accuracy and efficiency.

Traditionally, methods like grid search and random search have been used to find the best hyperparameters. Grid search systematically tests combinations of hyperparameters but can be slow and computationally expensive, especially as the number of hyperparameters increases. Random search picks hyperparameter values randomly and can sometimes find good settings faster than grid search, but it lacks consistency and can miss optimal settings.

As machine learning technology has evolved, more sophisticated techniques have been developed to tackle the limitations of traditional methods. Bayesian optimization, for example, uses statistical models to predict the performance of hyperparameters and focuses the search on areas likely to yield better results. This method has become popular for its efficiency and effectiveness in managing large hyperparameter spaces. [4]

Moreover, the advent of machine learning frameworks that support automatic hyperparameter tuning, such as Optuna and Hyperopt, has further advanced the field. These frameworks provide flexible and efficient tools that automate the trial-and-error process, using advanced algorithms to guide the search towards promising regions of the hyperparameter space.

Understanding these developments helps set the stage for introducing our approach with SMO, which builds upon these innovations to offer further improvements in hyperparameter optimization. As machine learning applications grow across various industries, the need for efficient and effective model tuning becomes increasingly important, underscoring the relevance of this research.

## III. ALGORITHMS

### *DATA PRE-PROCESSING*

The data preprocessing step is crucial for getting the dataset ready for further analysis. First, the script ensures that the CSV file exists and is in the correct format to confirm that the data can be processed accurately. As it goes through each line of the CSV file, the script uses a function to accurately interpret and transform each piece of data from string form into a more usable format. For numerical data, it converts strings to integers or floats, and it cleans up any non-numeric data by removing unnecessary spaces. This preprocessing is essential to make sure the data is clean and properly formatted, which is important for keeping the data manipulation and analysis reliable and accurate.

### *A. GRID SEARCH - HYPERPARAMETER OPTIMIZATION TECHNIQUE*

Grid search is a method used in machine learning to find the best settings for a model, particularly useful in models like Support Vector Regression (SVR). This method systematically tests every combination of settings within a specified range to determine which one performs the best. It does this by adjusting parameters such as the model's cost (C) and epsilon ($\epsilon$), which influence how the model is trained. These parameters are tested over a range from a very small number like 0.001 to a slightly larger number like 0.1, which allows us to explore their effects comprehensively.

For instance, adjusting the cost parameter (C) helps balance between achieving high accuracy and avoiding overfitting. A higher C value can push the model to better performance by making it more sensitive to the training data, which is beneficial for capturing complex patterns. Similarly, fine-tuning $\epsilon$ determines the width of the margin within which no penalty is given in the training loss function for errors. Properly setting epsilon can lead to a model that generalizes well on unseen data by not being overly sensitive to slight errors.

The process involves training the model with different combinations of these parameters and then testing it to see how accurately it can predict outcomes. The accuracy for each combination is measured using the Mean Absolute Error (MAE), which calculates the average error in predictions. Every result, including each pair of settings and their accuracy, is recorded meticulously. This comprehensive record helps in analyzing which settings led to the best performance of the model.

Additionally, the duration of the process is monitored to gauge the method's efficiency. The results are often displayed in a 3D graph to visually represent which combinations of settings yielded better accuracy. While grid search is comprehensive and ensures that all possible options are explored, it is known for requiring significant computational time and resources. This detailed approach is advantageous when the utmost accuracy of a model is crucial and there is a readiness to invest in the required resources.

Using grid search helps us carefully pick the best settings for our model. This method lets us try out different combinations of parameters in a structured way, so we can make sure we're making smart decisions to improve our model's accuracy. By doing this, we make our model not only more accurate but also better at handling different situations. This makes grid search a really useful strategy in machine learning projects.

The reason we use grid search is because it's been proven effective, and reliable and checks every possible combination of parameters. This is super important, especially in fields like healthcare or finance where having an accurate model is really crucial. It's commonly used in studies and textbooks as the go-to method for tuning parameters. Even though grid search needs a lot of computing power, its thorough and systematic approach is worth it. It's a reliable way to explore all the different settings for our model and find the best ones. Grid search serves as a solid benchmark for comparing newer, more complex techniques. Its thoroughness makes it easy to see if these new methods are truly better.

### *B. OPTUNA - HYPERPARAMETER OPTIMIZATION TECHNIQUE*

Optuna is a modern tool designed to fine-tune the settings, known as hyperparameters, of machine learning models. Its primary goal is to make it easier to find the best values for these settings, thereby improving the accuracy of the model's predictions. For example, Optuna can optimize settings in a Support Vector Regression (SVR) model, particularly focusing on two crucial parameters: the cost (C) and epsilon (e). These parameters are important because they greatly influence how well the model fits the data and makes predictions.

Optuna uses a systematic method to explore these parameters by conducting iterative trials, efficiently determining their optimal values. The framework of Optuna automatically adjusts and refines the search space based on the outcomes of previous trials, which helps in optimizing the predictive accuracy and generalization abilities of the SVR. This method allows Optuna to smartly and efficiently navigate through the parameter space, reducing the need for manual guesswork and saving computational resources.

Optuna enhances the search for the best parameter combinations by adopting a smart trial-and-error method. It starts with initial guesses and evaluates how effective they are. Based on these evaluations, Optuna decides on which settings to try next. This approach is more efficient than testing every possible combination, like in grid search, because it learns from past results and quickly zeroes in on better parameters.

The optimization process in Optuna is centered around what you define as the objective function. This involves training the SVR with a training dataset and then checking its accuracy on a validation set using the Mean Absolute Error

(MAE). MAE measures the average discrepancy between the predicted values and the actual values, with lower values indicating better accuracy. Each set of tested settings is called a trial, and Optuna continuously proposes new trials that refine these settings based on the outcomes of previous ones. This cycle repeats for a specified number of iterations, each aimed at enhancing the model's performance.

Optuna not only finds the best settings more quickly but also requires less computational power compared to methods that try every possible setting like grid search. This efficiency is especially beneficial for complex models where manual tuning is impractical or when computational resources are limited. Optuna's ability to adaptively learn from each trial's results makes it a powerful tool for improving both the accuracy and efficiency of machine learning models.

### C. SEQUENTIAL MODEL OPTIMIZATION - HYPERPARAMETER OPTIMIZATION TECHNIQUE

Sequential Model Optimization (SMO) offers a refined approach to enhancing predictive models through strategic evaluation of data points. Initially, SMO constructs two preliminary models—one representing the optimal or 'best' outcomes and the other representing the suboptimal or 'rest'. This setup is critical for identifying data points that lie on the boundary between these two models, thereby pinpointing those that are most informative for further evaluation[2].

In SMO, we pick data points that are closely contested between two models, which are considered the most important for further analysis. After choosing one of these points, we carefully assess it and use what we learn to update both models. We keep repeating this process, focusing on different key data points each time. By doing this, we aim to refine the boundary between what the models see as the 'best' and the 'rest'. This targeted approach makes sure that each analysis we do helps improve the accuracy of the models as much as possible, even though we're using limited resources.

In practical implementation, this system processes data from CSV files, feeding it into the SMO framework where it employs algorithms to compute metrics such as mean squared error. These metrics help determine the effectiveness of each iteration of the models. The system also incorporates a mechanism to strategically shuffle and select data points, optimizing the sequence of evaluations to focus on those predicted to yield the most valuable information. As a result, the models become increasingly accurate with each iteration, quickly converging to the maximum possible accuracy as possible.

This methodology proves particularly advantageous in scenarios where data evaluation is associated with high costs. By minimizing the number of necessary evaluations while maximizing the informational yield of each, SMO not only reduces costs and saves time but also enhances the overall efficiency and effectiveness of the modeling process.

## IV. METHODS

Below FIGURE 1 gives us a visualization of the characteristics of different datasets used. TABLE 1 gives us the summary of the dataset which includes the number of independent and dependent variables namely X and Y and number of rows present in a dataset.

### A. DATA

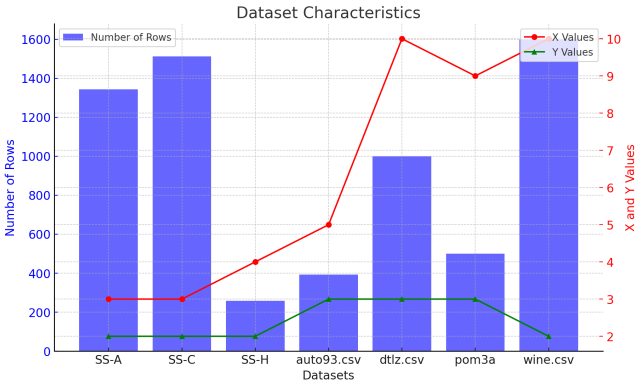


**FIGURE 1.** Dataset characteristics

**TABLE 1.** Summary of Datasets

| Datasets | X | Y | Rows |
|---|---|---|---|
| SS-A | 3 | 2 | 1343 |
| SS-C | 3 | 2 | 1512 |
| SS-H | 4 | 2 | 259 |
| auto93.csv | 5 | 3 | 392 |
| dtlz.csv | 10 | 3 | 1000 |
| pom3a | 9 | 3 | 500 |
| wine.csv | 10 | 2 | 1599 |

1) **`SS-A.csv`**: This dataset contains 1,343 entries detailing performance metrics for a data processing system, including settings such as `Spout_wait`, `Splitters`, and `Counters`. These settings impact system processing speed and delay, measured through metrics like `Throughput+` and `Latency-`. The data is essential for optimizing systems handling high data volumes, critical in sectors like telecommunications or financial services. The correlation matrix

**TABLE 2.** Correlation Matrix for the dependent variable in SS-A Dataset

| | Throughput+ | Latency- |
|---|---|---|
| **Throughput+** | 1.000000 | -0.603095 |
| **Latency-** | -0.603095 | 1.000000 |

for the SS-A dataset in TABLE 2 reveals a moderately strong negative correlation of -0.603095 between Throughput+ and Latency-. This suggests that as the throughput increases, the latency tends to decrease, and vice versa. This relationship indicates that higher throughput typically leads to better performance (lower latency), which is a desirable outcome in systems where efficient processing is critical. These findings are

significant as they provide insights into how improvements in throughput can potentially reduce latency in operational settings.

2) **SS-C.csv**: Consisting of 1,512 records, this dataset tracks the performance of various configurations in a data processing system. It features settings such as `Spout_wait`, `Splitters`, and `Counters`, affecting data processing speed (`Throughput+`) and delay (`Latency-`). The insights provided are crucial for optimizing high-volume data processing systems.
3) **SS-H.csv**: This dataset contains 259 records detailing the performance of hardware configurations in terms of energy usage and runtime efficiency. Each record has settings such as width, complexity, and FIFO buffer size, alongside metrics like `Energy-` and `Inv_runtime-`, where lower values indicate better performance. It's invaluable for optimizing hardware designs to reduce energy costs and enhance operational speed.
4) **auto93.csv**: With 392 records, this dataset documents 1970s automobiles, providing details such as the number of cylinders (`Clndrs`), engine displacement (`Volume`), horsepower (`HpX`), model year (`Model`), origin, vehicle weight (`Lbs-`), acceleration (`Acc+`), and fuel efficiency (`Mpg+`). The data is useful for analyzing historical automotive performance trends. The

**TABLE 3.** Correlation Matrix of the dependent variables in auto93 dataset

| | Lbs- | Acc+ | Mpg+ |
|---|---|---|---|
| **Lbs-** | 1.000000 | -0.416839 | -0.790639 |
| **Acc+** | -0.416839 | 1.000000 | 0.410348 |
| **Mpg+** | -0.790639 | 0.410348 | 1.000000 |

correlation matrix for vehicle performance characteristics in Table 3 shows that weight negatively impacts both acceleration and fuel efficiency, with a moderate negative correlation to acceleration (-0.416839) and a strong negative correlation to fuel efficiency (-0.790639). Conversely, there is a moderate positive correlation (0.410348) between acceleration and fuel efficiency, indicating that vehicles that accelerate faster tend to be more fuel-efficient. These relationships suggest that reducing vehicle weight could enhance both acceleration and fuel efficiency, reflecting the complex interplay between these key vehicle attributes.

5) **dtlz.csv**: The DTLZ datasets, including files from `dtlz2.csv` to `dtlz7.csv`, each feature 1,000 rows with ten input variables and three performance indicators, typically used for balancing multiple objectives simultaneously. These datasets are particularly useful for testing and improving decision-making algorithms in fields requiring the optimization of cost, efficiency, and performance. The DTLZ (Deb-Thiele-Laumanns-Zitzler) dataset comprises several test problems designed for multi-objective optimization. Each problem within the DTLZ series presents a unique set of characteristics, making them valuable for evaluating optimization algorithms.

DTLZ1, for instance, features a linear Pareto-optimal front, making it relatively easier to optimize compared to other problems in the series. However, as the number of objectives and decision variables increases, so does the complexity of the optimization task. DTLZ2 shares similarities with DTLZ1 but introduces a non-linear Pareto-optimal front, having a greater challenge for optimization algorithms. DTLZ3 offers a scalable number of objectives. This problem introduces non-linearity to the objective functions, providing a more complex landscape for optimization. DTLZ4 builds upon DTLZ3 by adding a distance-based scaling mechanism, aiming to spread solutions more evenly along the Pareto front and enhance diversity in the solution space. DTLZ5 and DTLZ6 further increases the complexity of the optimization problems. DTLZ5 introduces oscillatory Pareto-optimal front layers, adding challenges for maintaining a balance between convergence and diversity. DTLZ6 features a front shape, combining convex and concave segments, thereby increasing the optimization challenge. Lastly, DTLZ7 adds another layer of complexity by introducing non-linearity to the objective functions, akin to DTLZ3, but with a different Pareto-optimal front shape. Each DTLZ problem serves as a valuable benchmark for evaluating the performance of multi-objective optimization algorithms.

**TABLE 4.** Correlation Matrix of the dependent variables in DTLZ Dataset

| | Y0- | Y1- | Y2- |
|---|---|---|---|
| **Y0-** | 1.000000 | -0.023865 | -0.556336 |
| **Y1-** | -0.023865 | 1.000000 | -0.532331 |
| **Y2-** | -0.556336 | -0.532331 | 1.000000 |

The correlation matrix for the DTLZ2 dataset in TABLE 4 shows a minimal negative correlation between Y0- and Y1- (-0.023865), suggesting almost no linear relationship. There is a moderate negative correlation between Y0- and Y2- (-0.556336), indicating that increases in Y0- are moderately associated with decreases in Y2-. Y1- and Y2- also show a moderate negative correlation (-0.532331), implying a similar inverse relationship. These correlations suggest that changes in one variable tend to be associated with opposite changes in the others, but the relationships are not extremely strong, except between Y1- and Y2-.

6) **pom3a.csv**: This dataset, with 500 entries, focuses on project management metrics within software development. It includes parameters like `culture`, `criticality`, and `team size`, along with metrics such as `cost` and `success score`, crucial for refining project management strategies.
7) **wine.csv**: Containing 1,599 records, this dataset details the chemical properties and quality ratings of var-

ious wines, with attributes such as `acidity`, `sugar content`, and `alcohol percentage`. This information helps the wine industry enhance product quality by understanding how each chemical attribute impacts the overall rating. The correlation matrix for the Wine dataset in TABLE 5 indicates a positive correlation of 0.476166 between Alcohol- and Quality+. This moderate positive correlation suggests that wines with higher alcohol content tend to be associated with higher quality ratings. This relationship highlights a significant trend where alcohol level is one of the factors that can influence the perceived quality of wine.

**TABLE 5.** Correlation Matrix of the dependent variables in Wine Dataset

| | Alcohol- | Quality+ |
|---|---|---|
| **Alcohol-** | 1.000000 | 0.476166 |
| **Quality+** | 0.476166 | 1.000000 |

### B. EXPERIMENTAL SETUP

In the project, we focused on refining machine learning models to predict continuous values accurately, using methods like Support Vector Regression (SVR). These methods are particularly effective for tasks that require precise predictions, which are common in fields such as finance and scientific research.

We chose SVR because it handles complex, non-linear relationships between data points well. This capability comes from SVR's use of kernel functions, which help the model operate effectively even in high-dimensional spaces. This makes SVR ideal for complicated datasets where simple linear models might not perform well. The effectiveness of SVR for such tasks is well-documented in [6].

To optimize our models, we implemented both grid search and Optuna. Grid search is comprehensive, examining every combination of parameters to ensure the best possible model settings are identified. Although it's resource-intensive, its thoroughness makes it invaluable, especially when precision is critical.

We also incorporated Optuna, an advanced optimization framework that uses Bayesian optimization techniques. Optuna enhances the parameter tuning process by building a probabilistic model of the objective function. It intelligently predicts which parameter settings are likely to yield improvements, making the tuning process more efficient and effective.

Parameter settings for 'c' and 'epsilon' were chosen based on preliminary data analyses and existing research, suggesting that specific ranges might best balance model complexity with the ability to generalize well on new data.

Data pre-processing was another critical aspect of our setup. We normalized and scaled the features to ensure that all inputs to the models were treated equally. This step is crucial for models like SVR, which rely heavily on distance calculations between data points. Proper scaling ensures no single feature dominates the model's decision-making process.

Additionally, we introduced a unique metric called 'distance to heaven' to simplify how we assess model outputs. This metric measures how far each prediction is from an ideal state, making it easier to understand the model's effectiveness at a glance.

Based on the above assumptions we attempt to search through the hyper-parameter space as efficiently as possible, also we compare our results using mean square error on the baseline model i.e. grid search.

### C. EVALUATION METRICS

When assessing the accuracy of our machine learning models, we primarily rely on two metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE). MAE measures the average magnitude of errors in a set of predictions, without considering their direction. This metric is straightforward as it calculates the average over the test sample of the absolute differences between prediction and actual observation, where each difference is given equal weight. This feature makes MAE a preferred choice when it's crucial to understand the accuracy of predictions in straightforward, interpretable units similar to those of the target variable. It's especially valuable because it's less sensitive to outliers, meaning that a few very large errors won't disproportionately affect the average error.

MSE, meanwhile, calculates the average of the squares of the errors—that is, it takes the average squared difference between the estimated values and what is actually observed. This approach means that larger errors have a disproportionately large effect on the overall error metric, which can be useful when it's particularly important to penalize large deviations heavily. The squaring of prediction errors in MSE not only emphasizes larger errors but also facilitates the use of gradient-based optimization methods because it ensures that the error gradient is smooth and continuous.

In our experimental workflow, the model first learns from a designated set of data, known as the training set. Following the training phase, the model's predictions are tested against a new dataset, referred to as the test set. The accuracy of these predictions is then quantitatively evaluated by comparing the predicted outcomes against the actual outcomes using both MAE and MSE. This step is crucial as it provides a clear measure of how well the model performs under real-world conditions.

During the tuning phase, particularly when we adjust hyperparameters using methods like Grid Search, Random Search, or Optuna, the effectiveness of different sets of parameters is assessed based on the values of MAE and MSE. Configurations that result in the lowest values of these metrics are considered optimal, as they indicate the model's predictions are closest to the actual data points.

We selected MAE and MSE as our primary evaluation metrics because they offer a clear statistical basis to evaluate and compare the performance of various regression models effectively. They help quantify the extent to which the model

deviates from the true values, providing a direct indicator of prediction accuracy. This approach not only supports rigorous assessment but also ensures that the results are easily understandable and actionable. Moreover, their consistent application across different types of regression models and datasets makes them one of the best tools in our experimental toolkit.

### D. STATISTICAL METHODS

Significance testing is a key statistical tool used to determine if the results of a study are likely caused by something specific that the researchers are testing for, rather than by chance. This kind of testing is typically used to check the validity of a "null hypothesis," which states that there is no effect or no difference between the groups or conditions being tested. A key part of this testing is the "p-value," which is a number that tells us how likely it is that we would observe the study results if the null hypothesis were true. Generally, if the p-value is less than 0.05, we consider the results significant enough to reject the null hypothesis.

The Kruskal-Wallis test is a non-parametric method used when comparing more than two groups, especially when you can't assume that the data is normally distributed. This test looks at whether the medians of these groups are different by ranking all the data points and analyzing these ranks. A very low p-value from this test (like 3.44e-49) indicates that there is a statistically significant difference in at least one group's median compared to others. In some cases, such as with dataset SS-H, the p-value is close to 0.05, indicating that the difference between the groups might not be statistically significant.

Effect size testing, on the other hand, helps us understand not just if there is a statistically significant difference, but how large that difference is. One common measure of effect size is Cohen's d, which expresses the size of the difference in terms of standard deviations. This tells us about the practical significance of the difference—essentially, how much it matters in real-world terms. For example, in several datasets like dtlz2, dtlz4, and others, the effect sizes are large (Cohen's d values above 0.8), indicating substantial differences in how the two optimization methods affected the Mean Absolute Error (MAE). Conversely, datasets like auto93 and dtlz7 showed moderate effect sizes, indicating smaller, yet meaningful, differences between the methods.

In summary, the statistical tests used here help us understand whether different methods of optimization in machine learning models have a significant effect and how large these effects are, indicating the practical implications of choosing one method over another. This is crucial for improving model performance based on the optimization method used. TABLE 6 consists of significance and effect size test values on different datasets.

## V. RESULTS

Table 7 showcases the results of hyperparameter optimization for SVM models across several datasets, utilizing Grid Search, Optuna, and Sequential Model Optimization (SMO). For each dataset, the methods are evaluated based on the best Mean Squared Error (MSE), the hyperparameters that led to this MSE, and the computational time required. For instance, in the SS-C.csv dataset, Grid Search achieved an MSE of 0.0863 with parameters C=0.076 and $\epsilon$ = 0.044, requiring a significant time investment of 739.14 seconds. Optuna improved upon this with an MSE of 0.0774, using different parameters C = 0.084871, $\epsilon$ = 0.0091) and drastically reducing the time to 63.32 seconds. SMO outperformed both, achieving the lowest MSE of 0.0565 in only 48.43 seconds, although the specific parameters are not provided. This pattern of varying effectiveness and efficiency is consistent across other datasets like SS-H.csv, auto93.csv, pom3a.csv, and wine.csv, highlighting the strengths and weaknesses of each method in different contexts.

**TABLE 6.** Significance test and effect size test results

| Datasets | Kruskal-Wallis Test | Cohen's d |
|---|---|---|
| auto93 | $3.62 \times 10^{-18}$ | 0.840 |
| dtlz2 | $6.78 \times 10^{-32}$ | 1.356 |
| dtlz3 | $3.44 \times 10^{-49}$ | -1.947 |
| dtlz4 | $1.62 \times 10^{-28}$ | 1.208 |
| dtlz5 | $1.46 \times 10^{-31}$ | 1.337 |
| dtlz6 | $3.19 \times 10^{-24}$ | 1.219 |
| dtlz7 | $1.71 \times 10^{-28}$ | 0.697 |
| pom3a | $1.25 \times 10^{-6}$ | 0.605 |
| SS-A | $2.59 \times 10^{-51}$ | 1.340 |
| SS-C | $2.65 \times 10^{-30}$ | -0.691 |
| SS-H | $4.04 \times 10^{-2}$ | 0.1272 |
| wine | $6.02 \times 10^{-17}$ | 0.921 |

**TABLE 7.** Summary of Optimization Results Across Datasets

| Dataset | Method | Best MSE | Parameters | Time (s) |
|---|---|---|---|---|
| SS-C.csv | Grid | 0.0863 | c=0.076, e=0.044 | 739.14 |
| | Optuna | 0.0774 | c=0.084871, e=0.0091 | 63.32 |
| | SMO | 0.0565 | c=0.084, e=0.008 | 48.43 |
| SS-H.csv | Grid | 0.0498 | c=0.099, e=0.001 | 25.71 |
| | Optuna | 0.0232 | c=0.066, e=0.0117 | 11.33 |
| | SMO | 0.0481 | c=0.072, e=0.010 | 4.78 |
| auto93.csv | Grid | 0.0478 | c=0.095, e=0.010 | 48.05 |
| | Optuna | 0.0340 | c=0.092046, e=0.037 | 12.60 |
| | SMO | 0.0848 | c=0.099, e=0.04 | 8.56 |
| pom3a.csv | Grid | 0.0807 | c=0.099, e=0.065 | 111.76 |
| | Optuna | 0.0693 | c=0.080609, e=0.069 | 16.64 |
| | SMO | 0.0848 | c=0.081, e=0.054 | 8.56 |
| wine.csv | Grid | 0.0639 | c=0.096, e=0.001 | 770.79 |
| | Optuna | 0.0537 | c=0.099959, e=0.0034 | 77.32 |
| | SMO | 0.0631 | c=0.0998, e=0.002 | 53.23 |

**Research Question 1: What is the impact of initial hyperparameter selection on the effectiveness of the SVM model?** Hyper-parameters define how an SVM model performs on a certain dataset. They also dictate if the model converges faster on a particular dataset. Depending on the initial hyper-parameters used for a model the model may find the global Optimia (minima for errors) sooner.

**Research Question 2: How does the performance of Grid Search compare to more advanced techniques like Optuna and Sequential Model Optimization (SMO) in**

**terms of computational efficiency and accuracy?** When comparing the techniques across various datasets, Grid Search generally consumed the most time and often did not achieve the best MSE values. For example, in the auto93.csv dataset, Grid Search yielded an MSE of 0.0478 in 48.05 seconds, whereas Optuna and SMO performed better or comparably in significantly less time. Notably, Optuna outperformed Grid Search with an MSE of 0.0340 in just 12.60 seconds. SMO did not always outperform other methods but typically required the least computation time.

**The findings demonstrate that advanced optimization methods like Optuna and SMO are more efficient than Grid Search, both in terms of computational time and effectiveness in finding optimal hyperparameter settings. These methods are particularly beneficial for large-scale applications where computational resources and time are constraints.**

**Research Question 3: What are the limitations of using a model like SMO for SVM regression (hyperparameter tuning), and how do these limitations affect the optimization outcomes?** The application of SMO in SVM regression shows promising results but also reveals certain limitations, particularly in the representativeness of the dataset samples used in the modeling process. If the subset of data does not capture the complete variability of the dataset, the predictions could be biased, leading to suboptimal hyperparameter settings. For instance, while SMO typically required less computational time, its performance was not consistently superior in terms of MSE, as seen in the auto93.csv and pom3a.csv datasets. **Overall, while SMO can significantly reduce computational loads, its effectiveness is constrained by the representativeness of the data samples used. This can lead to variations in model performance, highlighting the importance of careful data selection and pre-processing in the application of SMO techniques.**

## VI. DISCUSSION

### A. WHAT WAS LEARNED

The comprehensive evaluation of hyperparameter optimization methods in this study has revealed several key insights into the performance and efficiency of SVM regression models. Firstly, the importance of initial hyperparameter selection was clearly demonstrated, with different techniques showing substantial variance in both effectiveness and computational demands. Optuna and Sequential Model Optimization (SMO) often provided superior performance compared to traditional Grid Search, highlighting the advancements in optimization technology that can significantly enhance model-tuning processes.

A surprising finding was the variability in the performance of SMO; while it was expected to consistently outperform other methods due to its advanced model-based strategy, its effectiveness was highly dependent on the representativeness of the data samples used. This points to a nuanced understanding that no single method universally outperforms others across all scenarios. The selection of the optimization technique must consider specific dataset characteristics and the practical constraints of the modeling environment.

### B. THREATS TO VALIDITY

Several factors pose threats to the validity of our findings:

1) **Representativeness of Data:** The effectiveness of SMO hinges on the assumption that the subset of data used is representative of the entire dataset. Any deviation from this can lead to biased results and suboptimal hyperparameter settings, thereby affecting the generalizability of the optimization outcomes. To reduce the dependence on the initial subset taken into consideration we have run SMO 20 times and averaged the results.
2) **Generalization Across Datasets:** The datasets used in this study, while diverse, may not cover all types of scenarios encountered in different applications of SVM regression. This limits the applicability of our conclusions to similar types of data and model configurations.
3) **Methodological Biases:** The inherent biases in the experimental design, such as the choice of datasets, the range of hyperparameters considered, and the evaluation metrics used, could also impact the findings. These choices may favor certain optimization methods over others, skewing the results.

### C. FUTURE WORK

Based on our findings, several directions are recommended for future research:

1) **Hybrid Optimization Techniques:** Developing hybrid models that combine the robustness of Grid Search with the efficiency of Optuna and the intelligence of SMO could potentially overcome the limitations of individual methods. Such models would harness the strengths of each technique to provide a more balanced and universally applicable solution.
2) **Broader Application and Testing:** Future studies should aim to test these optimization methods across a wider range of datasets and in different machine-learning contexts. This would help in validating the generalizability of the observed trends and in refining the techniques for broader application.
3) **Advanced Statistical Analysis:** Employing more rigorous statistical methods to assess the impact of hyperparameter optimization on model performance could provide deeper insights. Techniques such as cross-validation, bootstrap methods, and Bayesian statistical models could be explored to provide a more robust evaluation of method efficacy.
4) **Real-World Implementations:** Applying these optimization techniques in real-world scenarios and assessing their practical implications in live environments would offer valuable perspectives on their operational effectiveness and scalability.

## VII. CONCLUSION

The paper effectively addresses the challenge of optimizing hyperparameters in Support Vector Machine (SVM) regression models, focusing on the penalty parameter C and the slack variable $\epsilon$. Through comparative analysis, the paper evaluates three different optimization methods: grid search, Optuna, and Sequential Model Optimization (SMO), each with distinct approaches to fine-tuning the SVM's parameters.

As detailed in the results, optimization techniques like Optuna and SMO fare better when compared with baseline models like grid search. SMOs depend heavily on the initial 'lite' dataset, each point in the 'dark' dataset sequentially taken which produces the maximum in the acoustic function. This is how SMO builds point by point. At the end of SMO data present in the 'lite' dataset has the most variance thereby confirming our main hypotheses that a finite amount of subset of the total data can work remarkably well.

The research demonstrates that Optuna and SMO, leveraging more advanced and dynamic strategies, outperform the traditional grid search method in terms of computational efficiency and model accuracy. Optuna uses Bayesian optimization to prioritize more promising regions of the hyperparameter space, thus speeding up the search process. SMO introduces an innovative model-based approach, which reduces computational demands by using only a subset of data for initial evaluations.

However, the study also acknowledges the limitations associated with these methods, particularly the reliance of SMO on initial data subsets that may not always represent the broader data characteristics fully. This could lead to biased or inaccurate model predictions if the subset is not sufficiently representative.

The paper's contributions are significant in integrating various optimization techniques and demonstrating through empirical evidence that modern methods can enhance the efficiency and efficacy of hyperparameter tuning in SVM models. The research underscores the potential of these advanced methods to transform the landscape of machine learning model optimization, particularly in real-world applications where computational resources and time are at a premium.

In conclusion, the research provides a robust comparative analysis and offers a clear path forward for improving SVM performance through intelligent hyperparameter optimization. It invites further investigation into hybrid models that might combine the strengths of these methods to achieve even greater efficiencies and accuracies in machine learning applications.

.

## REFERENCES


[1] Amritanshu Agrawal, Tim Menzies, Is "Better Data" Better Than "Better Data Miners"? Raleigh, NC, USA https://arxiv.org/pdf/1705.03697.pdf

[2] Tianpei Xia, Rui Shu, Xipeng Shen, Tim Menzies.ISequential Model Optimization for Software Effort Estimation Raleigh, NC, USA https://arxiv.org/abs/1912.04189

[3] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. Advances in Neural Information Processing Systems, 25. https://arxiv.org/abs/1206.2944

[4] [Shahriarietal.,2016] https://ieeexplore.ieee.org/document/7352306

[5] Bergstra and Bengio's study, "Random Search for Hyper-Parameter Optimization" (Journal of Machine Learning Research, 2012). https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

[6] Harris Drucker, Linda Kaufman, "Support Vector Regressor Machines," Bell labs and Monmouth university, 1996. NeurIPS Proceedings.

[7] Hutteretal.,2011 https://ml.informatik.uni-freiburg.de/wp-content/uploads/papers/11-LION5-SMAC.pdf

[8] DTLZ dataset https://pymoo.org/problems/many/dtlz.html

[9] Wine quality dataset https://rb.gy/t2iyz1

[10] auto93 dataset: auto93.csv https://github.com/timm/lo/blob/main/data/auto93.csv

[11] pom dataset: http://multicomp.cs.cmu.edu/resources/pom-dataset/

[12] SS dataset: https://arxiv.org/pdf/1801.02175.pdf