Data-driven design of a supercompressible material

ENGN 2350: Data-Driven Design and Analysis of Structures and Materials
Professor Miguel Bessa

Mohammad Majidi, Sungwon La

December 9, 2023

1 Data Characterization

In order to design a new super-compressible meta-material with optimal geometrical and material properties, machine learning will be used with data from FEA simulations in ABAQUS. The geometry of the material is defined by the top and bottom diameters D_1 and D_2, the height P, and the cross-section parameters of the vertical longerons: the cross-sectional area A, moments of inertial I_x and I_y, and torsional constant J. The mechanical properties of the material are defined by the Young's modulus E and shear modulus G. These are the features that must be optimized to yield the best performance metrics of coilability, critical buckling stress, and absorbed energy.

Data has been created by running ABAQUS simulations on the material with varying features. The three output values of the simulations are coilability, critical buckling stress σ_{crit}, and absorbed energy E_{abs}. Two different datasets are provided, with the input design space sampled using Sobol sequence:

1. a dataset with 50000 experiments that is parametrized by 7 parameters: $\frac{D_1-D_2}{D_1}$, $\frac{P}{D_1}$, $\frac{I_x}{D_1^4}$, $\frac{I_y}{D_1^4}$, $\frac{J}{D_1^4}$, $\frac{A}{D_1^2}$, $\frac{G}{E}$
2. a simplied dataset with 1000 experiments that is parametrized by 3 parameters: $\frac{d}{D_1}$, $\frac{D_2-D_1}{D_1}$, $\frac{P}{D_1}$

1.1 Bounds of each input variable

The bounds of the input variables for the 3D dataset are:

ratio-d	(0.004, 0.072933)
ratio-pitch	(0.25, 1.498779)
ratio-top-diameter	(0, 0.799219)

The bounds of the input variables for the 7D dataset are:

ratio-area	(1.17e-05, 0.0041)
ratio-Ixx	(1.128e-11, 0.000001)
ratio-Iyy	(1.128e-11, 0.000001)
ratio-J	(1.353e-11, 0.000008)
ratio-pitch	(2.5e-01, 1.499981)
ratio-top-diameter	(0, 0.799976)
ratio-shear-modulus	(3.5e-02, 0.449987)

1.2 Plotting Combinations of the Input Features

The combinations of every pair of input features for the 3-D dataset (limited to the first 100 points of the dataset) are plotted in a scatter plot inn Fig. 1, and a 3D scatterplot with all three features is plotted in Fig. 2. It appears from the plots that the combination samples are regularly distributed, covering the entire space of possible combinations of the input parameters in an even manner. Generally, for randomly-generated sampling sequences, clustering and irregularities can commonly be seen. However, the Sobol sequence is a quasi-random method that focuses on evenly distributing the points without clustering and irregularity. The combinations of the input variables are thus vary uniform in their distribution.

Table 1: Number of Points for Each Coilable Value in the 3D dataset

Coilable	Number of Points
0	318
1	214
2	468

Table 2: Number of Points for Each Coilable Value in the 7D dataset

Coilable	Number of Points
0	17669
1	32331

1.3 Histograms of the Output Variables

Histograms of the output variables, coilable, σ_{crit} and E_{abs} for the 3D dataset, are shown in Fig. 3. The distribution of the output value of coilability for the 3D dataset is given in Table. 1. Similarly, histograms of the output variables for the 7D dataset are shown in Fig. 3, and the distribution of the output value of coilability for the 7D dataset is given in Table. 2.

Both of these datasets need to be pre-processed, as some data points have missing output values due to unsuccessful simulations, and some data points are outliers. For the missing output values of σ_{crit} and E_{abs}, imputation was used to replace the missing data with substituted values. Additionally, to remove the outliers, modified z-score method was used to remove the data more than 3 standard deviations away from the mean. For the 3D dataset, 27 outlier points have been removed. For the 7D dataset, 1285 outlier points have been removed. After pre-processing, the number of available points for the 3D dataset is 973 points, and the number of available points for the 7D dataset is 48715 points.

1.4 Finding the Optimal Points from the 3D Dataset

From examination of the original 3D dataset provided, the points corresponding to a reversibly coilable material with the maximum critical buckling stress σ_{crit} and the maximum energy absorption capability E_{abs} (before removal of outliers and imputation) are found and shown in Table 3. Note that the points with NaN as one of the outputs are not considered.

2 Finding Good Machine Learning Models

To train the machine learning models on the datasets, both the 3D and 7D datasets were split into training and test sets, with 75 percent of the data used for training. The 3D dataset used 729 points for training and 244 points for testing, while the 7D dataset used 36536 points for training and 12179 points for testing. The input dataset was scaled using Scikit-Learn's StandardScaler. The output dataset for coilability (classification) was not scaled, while the output dataset for σ_{crit} and E_{abs} were also scaled using the StandardScaler.

Table 3: Features and Outputs of Reversibly Coilable Material with the Maximum σ_{crit} and the Maximum E_{abs} in the 3D Dataset (before pre-processing)

Features	Maximum σ_{crit}	Maximum E_{abs}
ratio-pitch	0.278076	0.320801
ratio-d	0.035063	0.033783
ratio-top-diameter	0.724219	0.757812
coilable	1	1
sigma-crit	6.803301	6.735183
energy	3.635225	3.732142

2.1 Training with the 3D Dataset

2.1.1 Ternary Classification

The C-Support Vector Classifier, Decision Tree Classifier, and Neural Network Classifier were used to fit the data, with the coilability as the output with three possible values: not coilable (0), coilable but yields (1), and coilable (2). These models were selected because of their different respective strengths in classification tasks. The C-Support Vector Classifier excels in handling complex decision boundaries, as it can utilize kernel functions to handle nonlinear patterns in the data. The Decision Tree Classifier is computationally efficient with small datasets, and can naturally handle nonlinearity even without kernels or neurons. The Neural Network Classifier is excellent in handling complexity and nonlinearity, with its robust selection of hyperparameters. However, these models all have their unique weaknesses as well. For example, the SVC model does not scale well with large datasets, and it was found to be the most computationally expensive when actually running the code. The Decision Tree model is prone to overfitting, and is thus not good with handling noise, as it may try to fit the noise as if it were handling patterns in the data. The Neural Network model is very computationally expensive if it has many layers and many neurons, and creates a need for extensive hyperparameter tuning as the number of layers and neurons increases

Support Vector Classifier (SVC)

An SVC model with a radial basis function (RBF) kernel was created. The model was trained on the training data (`input_data_3D_train` and `output_data_3D_train['coilable']`), and predictions were made on the test set (`input_data_3D_test`). Accuracy scores were calculated using `accuracy_score`. Different types of kernels were used, including the linear kernel, the third-degree polynomial kernel, the sigmoid kernel, and the RBF kernel. Out of these kernels, the RBF kernel was found to have the greatest accuracy. Thus, the hyperparameters were set as follows:

`C`: 1.0 (regularization parameter)
`kernel`: 'rbf' (radial basis function kernel)
`gamma`: 'scale' (kernel coefficient).

Decision Tree Classifier (DT)

A Decision Tree Classifier was instantiated and trained on the training data. This model gets its name from the tree-like structure of its algorithm. The model recursively splits the dataset based on features and creates decision nodes. When a stopping condition (such as a maximum depth limit of recursive iterations, or a minimum number of samples at a node) is met, a leaf node is created, which represents the final classification for the data point. The ”gini” function was used to measure the quality of the split, and the ”best” splitting strategy was used to choose the best split.

Neural Network Classifier (NN)

An Neural Network Classifier was created and trained. The Multi-Layer Perceptron classifier was used, which optimizes the log-loss function using an optimizer of choice. The adam optimizer was used for this specific case, as adam is known to work well on large datasets, which will be useful for the 7D dataset. Predictions are made on the test set, and the hyperparameters were set as follows:

`hidden_layer_sizes`: (100,) (one hidden layer with 100 neurons).
`activation`: 'relu' (rectified linear unit activation function).
`solver`: 'adam' (optimization algorithm).
`alpha`: 0.0001 (L2 penalty).
`learning_rate`: 'constant' (learning rate schedule).
`max_iter`: 200 (maximum number of iterations).

Classifier	Accuracy
SVM	0.84
Decision Tree	0.80
NN	0.83

Table 4: Classification Accuracy Scores for the 3D Dataset

In the ternary classification of the 3D dataset, three classifiers—Support Vector Machine (SVM), Decision Tree, and Neural Network (NN)—were evaluated using confusion matrices, Fig. 5, and accuracy scores, Table 4. As can be seen, the confusion matrices illustrate the distribution of true positive, true negative, false positive, and false negative classifications for each classifier across the three classes. In interpreting the confusion matrices, Support Vector Machine (SVM) exhibits a higher accuracy of 0.84, with strong diagonal values across all classes. SVM effectively distinguishes between the three classes, while Decision Tree, with an accuracy of 0.80, shows comparatively lower correct classifications. Neural Network (NN) performs similarly to SVM, with an accuracy of 0.83. Both SVM and NN display notable success in correctly classifying instances across the three classes, emphasizing their effectiveness in the 3D dataset classification task.

Classifying with Two categories (coilable and not-coilable)

This process was repeated, but with the output values for coilability reduced to not coilable (0) and coilable (1). All of the output values with coilable but yields were reclassified to coilable. The classification results for the 3D dataset using Support Vector Machine (SVM), Decision Tree (DT), and Neural Network (NN) classifiers are presented in Table. 5. Overall accuracy is comparable between SVM and NN, both achieving 92%, while DT slightly lags behind at 89%. In terms of precision, SVM and NN demonstrate similar performance at 92%, while DT exhibits a slightly lower precision score of 88%. The recall scores indicate that SVM has the highest sensitivity at 96%, followed closely by DT and NN at 95%. Additionally, the F1 scores show a balanced performance across classifiers, with SVM and NN achieving 94%, and DT slightly lower at 92%. In summary, all three classifiers perform well, with subtle differences in precision and recall, highlighting the nuanced trade-offs between the models in handling the 3D dataset.

Model	Accuracy	Precision	Recall	F1 Score
NN	0.93	0.93	0.96	0.95
DT	0.92	0.93	0.95	0.94
SVM	0.92	0.93	0.95	0.94

Table 5: Classification Metrics for the 3D Dataset

In the classification of the 3D dataset, all three models—Neural Network (NN), Decision Tree (DT), and Support Vector Machine (SVM)—demonstrated strong performance, achieving accuracy scores of 92% to 93%. The precision, recall, and F1 scores were consistent across models, reflecting a balanced ability to correctly classify both positive and negative instances. The confusion matrix, shown in Fig. 6, further illustrates the proficiency of all models, with a notable number of true positives (122) and true negatives (58). While the differences in metrics among models are marginal, Neural Network slightly outperforms others in recall at 96%, indicating its effectiveness in capturing actual positive instances. These results collectively affirm the models' reliability in classification tasks on the 3D dataset when it has two categories. Table 6 reports the elapsed time for these three classifiers for both binary and ternary classification on the 3D dataset. With a limited dataset size of 1000 points, the observed runtimes for all models are relatively fast. The modest changes in elapsed time suggest that the models demonstrate efficiency, particularly considering the small dataset size.

Model	SVC	Decision Tree	Neural Network
Ternary	0.0128s	0.0053s	0.7325s
Binary	0.0148s	0.0035s	1.0879s

Table 6: Elapsed Time for Different Models on 3D Dataset

Contours of ternary classification

The classification results of the 3D dataset using the Support Vector Classifier (SVC), Gaussian Process, and Neural Network classifiers are presented in Figures 7, 8, and 9, respectively, where testing points are represented as circles. Upon examining these contours, it becomes evident that the boundaries predicted by SVC and NN exhibit remarkable accuracy in delineating each coilability region. Furthermore, these classifiers demonstrate the ability to generalize the overall pattern effectively, resulting in high predictive accuracy. However, in the case of the decision tree, its proficiency lies mainly in classifying training points. Unfortunately, the pattern it derives lacks the capacity for effective generalization to testing points, highlighting a limitation in its predictive capability.

2.1.2 Regression

Gaussian Process Regression, Random forest, and Ridge Regression were used to fit the data and predict the critical buckling stress σ_{crit} and absorbed energy E_{abs}. These models were selected because:

1. **Gaussian Process Regression:** Gaussian Process Regression is selected for its ability to model non-linear relationships and provide uncertainty estimates in predictions. The chosen kernel for the Gaussian process is the Matern kernel with a smoothness parameter (ν) set to 2.5. This kernel choice influences the flexibility of the model, with higher ν values leading to smoother functions and lower values allowing for more intricate, non-linear patterns in the predictions.

2. **Ridge Regression:** Ridge Regression is chosen due to its effectiveness in handling multicollinearity in the data. The regularization parameter, α, is set to 1.0, striking a balance between fitting the data well and preventing overfitting. This parameter controls the strength of the regularization, influencing the trade-off between model complexity and generalization.

3. **Random Forest Regression:** Random Forest is employed for its capability to capture complex relationships in the data and mitigate overfitting. The model utilizes an ensemble of 100 decision trees, specified by the 'n_estimators' hyperparameter. This ensemble approach helps enhance predictive accuracy by aggregating the predictions of multiple trees, providing robust estimates for the σ_{crit} and E_{abs}.

These models with their respective hyperparameters are tailored to address different aspects of the data, offering a diverse and comprehensive approach to predicting critical buckling stress and absorbed energy. The combination of Ridge Regression for regularization, Random Forest for ensemble learning, and Gaussian Process Regression for non-linear flexibility provides a robust framework for accurate and nuanced predictions in the context of critical buckling stress and abosorbed energy.

To determine which model works best, we analyzed the regression metrics for the Critical Buckling Stress (σ_{crit}) and Energy (E_{abs}), shown in Table. 7. As can be noted, Gaussian Process consistently outperforms the other models for both Critical Buckling Stress and Energy metrics. It has the lowest Mean Squared Error (MSE) values and high (R^2) values, indicating a better fit to the data. Random Forest also performs well, but slightly less effectively than GP. It has higher MSE and slightly lower R^2 values compared to GP. Ridge performs the poorest among the three models. It has significantly higher MSE values and lower R^2 values, suggesting a less accurate fit to the data.

Table 7: Regression Metrics for 3D dataset

Model	Metric	Critical Buckling Stress (σ_{crit})	Energy (E_{abs})
Gaussian Process	MSE	17.03	3.84
	R^2	0.96	0.98
Random Forest	MSE	22.84	6.42
	R^2	0.95	0.96
Ridge	MSE	116.98	39.20
	R^2	0.74	0.75

Reasoning:

- **Gaussian Process (GP)** is likely performing well because it is a flexible, non-parametric model that can capture complex relationships in the data. This is particularly beneficial in scenarios where the underlying relationship between features and the target variable is nonlinear or has intricate patterns.
- **Random Forest (RF)**, an ensemble learning method, is also effective but might not capture complex relationships as well as Gaussian Process. However, it still provides robust performance.
- **Ridge**, being a linear regression model, may struggle to capture the complexity of the underlying relationships in the data, leading to inferior performance.

In conclusion, based on the obtained metrics, the Gaussian Process model appears to be the most suitable for this regression task, offering the lowest errors and the highest coefficient of determination (R^2). Table 8 presents the elapsed time for Gaussian

Model	Sigma (3D)	Energy (3D)
GP	0.929s	0.756s
RF	0.403s	0.357s
RR	0.0044s	0.003s

Table 8: Elapsed Time for Regression Models on 3D Dataset

Process Regression (GPR), Random Forest Regression (RFR), and Ridge Regression (RR) on both the sigma and energy dimensions of the 3D dataset. The runtime results are as follows:

The observed differences in runtime among the regression models can be attributed to their inherent algorithmic complexities and the nature of the dataset. Ridge Regression (RR) demonstrates exceptional speed, which can be attributed to its straightforward closed-form solution that involves matrix operations, making it computationally efficient for this 3D dataset. On the other hand, Gaussian Process Regression (GPR) and Random Forest Regression (RFR) involve more complex processes, such as kernel computations and ensemble learning, respectively, which contribute to longer elapsed times.

It's important to note that the observed trends in runtime are specific to the current dataset. Different datasets or variations in dataset characteristics could result in varying performance characteristics for these regression models. The efficient performance of Ridge Regression, especially in this context, underscores its suitability for tasks with such characteristics.

The contours for the predicted σ_{cr} and E_{abs} using Gaussian process, Random Forest, and Ridge regression are shown in Figs. 10-12, respectively.

2.2 Training with the 7D Dataset

2.2.1 Classification

The C-Support Vector Classifier, Decision Tree, and Neural Network Classifier were once again employed to fit the 7D data, which comprises only two categories (coilable = 0 and 1). The same hyperparameters were utilized as in the 3D dataset. The corresponding metrics are presented in Table 7. Confusion matrices are shown in Fig. 13. As can be noted, in this binary classification, the Neural Network outperforms both SVM and the Decision Tree with the highest accuracy (97%), precision (97%), recall (98%), and F1 score (97%). SVM exhibits strong recall (98%) but slightly lower precision (96%), while the Decision Tree lags behind with lower values in accuracy (91%), precision (93%), recall (92%), and F1 score (93%). The Neural Network demonstrates superior performance in minimizing both false positives and false negatives, indicating a better balance between precision and recall. The complex, non-linear relationships captured by the Neural Network architecture contribute to its effectiveness in handling the intricate patterns within the data, making it the preferred choice for this specific binary classification task.

Metric/Classifier	SVM	Decision Tree	Neural Network
Accuracy	0.96	0.91	0.97
Precision	0.96	0.93	0.97
Recall	0.98	0.92	0.98
F1 Score	0.97	0.93	0.97

Table 9: Performance Metrics for SVM, Decision Tree, and Neural Network

Model	SVC	Decision Tree	Neural Network
7D, 2 Classes	15.951s	0.7445s	32.923s

Table 10: Elapsed Time for Classification Models on 7D Dataset (2 categories)

The increase in elapsed time from the 3D to the 7D dataset is substantial for all three models, shown in Table 10. This significant difference can be attributed to the higher dimensionality of the 7D dataset, which results in increased computational complexity during training. The Support Vector Classifier (SVC) and Neural Network (NN) particularly demonstrate a considerable rise in training time, highlighting the challenges posed by higher-dimensional data. The Decision Tree (DT) remains relatively efficient even with the higher dimensionality, showcasing its resilience to such increases. This efficiency is attributed to the inherently parallel and recursive nature of decision tree algorithms, which allows them to adapt well to an increased number of dimensions. The DT's ability to efficiently handle higher-dimensional datasets contributes to its speed in the 7D classification task. In summary, these findings underscore the impact of dataset dimensionality on the computational demands of classification models.

2.2.2 Regression

Neural Network, Ridge Regression, and Random Forest were used to fit the 7D dataset and predict the critical buckling stress (σ_{crit}) and absorbed energy (E_{abs}).

The initial intention was to perform regression analysis on a 7D dataset using Gaussian Processes (GP). However, due to technical challenges—specifically, a kernel failure during the fitting process—we opted to utilize a Neural Network (NN) instead. This decision was made to ensure the successful completion of the regression analysis, considering the constraints encountered with Gaussian Processes. The results are given in Table 11.

Model	Response Variable	MSE	R^2
Neural Network	σ_{crit}	59.53	0.93
Random Forest	σ_{crit}	54.21	0.93
Ridge Regression	σ_{crit}	405.06	0.50
Neural Network	E_{abs}	77305.29	0.59
Random Forest	E_{abs}	76457.67	0.59
Ridge Regression	E_{abs}	135987.31	0.28

Table 11: Regression Metrics for Predicting σ_{crit} and E_{abs} in the 7D Dataset

For the prediction of σ_{crit}, both Random Forest and Neural Network demonstrated robust performance with low MSE (54.21 and 59.53, respectively) and high R^2 values (0.93 for both). Ridge Regression, while achieving a moderate R^2 of 0.50, displayed a higher MSE (405.06) compared to the other models.

However, when predicting energy (E_{abs}), all three models faced substantial challenges, evidenced by notably high MSE values. The R^2 value of 0.59 across all three models indicates that only around 59% of the variance in energy predictions is explained by the models. An R^2 of 0.50 for Ridge Regression underscores the difficulty in capturing the underlying patterns in the data for this specific response variable.

The relatively poor performance of Ridge Regression may be attributed to its inherent constraint on coefficient magnitudes. Ridge Regression introduces a penalty term to the linear regression algorithm, discouraging large coefficients and promoting a more stable model. However, in scenarios where the underlying relationship is complex or involves strong interactions, Ridge Regression's regularization may oversimplify the model, resulting in suboptimal predictive performance. This can be especially pronounced in cases where the relationship between predictors and responses is intricate, as seems to be the case for both σ_{crit} and E_{abs} in the 7D dataset.

Additionally, the poor performance in accuracy in the energy when compared to the critical buckling stress can be attributed to the difference in the noisiness of the output variables. Critical buckling stress is deterministic, whereas energy has an element of stochasticity to it, and inherently has noise. The models may have overfit to the noisy data, yielding in poor performance for the predictions; additionally, the imputation performed was also done with nearest neighbors, which is a noise-inducing method, so the data processing also added noise, exacerbating the overfitting of the model.

Table. 12 shows the elapsed time for regression models on the 7D dataset. The results demonstrate variations in elapsed time among the regression models for both the sigma and energy dimensions in the 7D dataset. Random Forest Regression (RFR) tends to have longer runtimes, possibly due to the increased complexity associated with higher-dimensional data. Ridge Regression (RR) shows exceptional speed, consistent with its closed-form solution that efficiently handles the additional dimensions. Neural Network (NN) takes a moderate amount of time, serving as a replacement for Gaussian Process Regression (GPR) due to computational constraints. These findings highlight the diverse performance characteristics of regression models in the context of higher-dimensional datasets.

Model	σ_{crit}	E_{abs}
Random Forest Regression (RFR)	39.843s	48.016s
Ridge Regression (RR)	0.0078s	0.0106s
Neural Network (NN)	17.465s	23.840s

Table 12: Elapsed Time for Regression Models on 7D Dataset

2.3 Comparison of the Best Solutions for the 3D and 7D Dataset

Comparing the best solutions found for the 3D and 7D problems reveals insights into the scalability of the algorithms employed. In the 3D problem, all three classifiers—Support Vector Machine (SVM), Decision Tree, and Neural Network (NN)—demonstrated strong performance with accuracy scores ranging from 0.80 to 0.84. The confusion matrices further highlighted their proficiency in correctly classifying instances across three categories.

However, the transition to the 7D problem, particularly in predicting absorbed energy (E_{abs}), witnessed a noticeable drop in accuracy for all classifiers. The accuracies for E_{abs} regression models, including Random Forest and Neural Network, decreased, emphasizing the inherent challenges posed by higher-dimensional datasets. While the classifiers showcased resilience in maintaining accuracy for critical buckling stress (σ_{crit}), the drop in accuracy for E_{abs} highlights the increased complexity and potential limitations in accurately predicting responses in a higher-dimensional space.

These results underscore the importance of considering the impact of dimensionality on model performance and the need for sophisticated approaches to maintain accuracy in more complex datasets. Scalability concerns, especially in the context of predicting energy-related parameters, warrant further investigation into model refinement or alternative techniques to ensure accurate predictions in higher-dimensional spaces.

3 Influence of Training Points and Hyperparameters on the Machine Learning Models

3.1 Influence of the Number of Training Points

The effect of the number of training points on the C-Support Vector Classifier and the Gaussian Process Regressor was examined. Shown in Figure 14 are a table and plot of accuracy vs. the number of training points used by the SVM model, with different models using different values for the hyperparameters of C. It is evident that a clear trend exists that as the number of training points increase, the accuracy increases for the SVM model, regardless of what value of C was used. The only exception to this is for the limit of a very small number of training points with a large value for the regularization hyperparameter C, as shown in the dip in the plots for C values of 20 and 50. This is because a larger value for C leads to a stricter model that overfits on the training points. The decision boundary becomes overfit to the training dataset, which is less likely to be representative of the test dataset or the entire dataset with a smaller number of training points, resulting in a less accurate model. In addition, this indicates that the imputation performed in the data processing step may have made our dataset worse, as the noise induced by the imputation (nearest neighbors method) may have led to worse overfitting, leading to reduced accuracy with large C values for small datasets.

Shown in Figures 15 and 16 are a table and a plots of the R-Squared and MSE error metrics vs. the number of training points used by the GP model. The GP model used the Matern kernel, and the smoothness hyperparameter value was varied between the standard values used in Scikit-Learn of 0.5, 1.5, and 2.5. For less smooth function approximations corresponding to the values of 0.5 and 1.5, there is a consistent trend in that an increasing number of data points leads to greater accuracy. For the smooth function approximation corresponding to the value of 2.5, however, there is a certain point at which increasing the number of data points starts leading to less accuracy. A possible explanation for this is that as the number of training points increases, the GP model becomes more complex and overfits to the noise in the training data. This is critical if the underlying function representing the data happens to not be smooth, since the smoothness hyperparameter set to 2.5 assumes a very smooth function, and as a result, the discrepancy that occurs when attemtping to model a not-so-smooth function with a smooth function approximation can lead to a reduction in accuracy when testing other points.

3.2 Influence of the Hyperparameters

The effects of the number of the C hyperparameter on the C-Support Vector Classifier on the accuracy of the model was examined, and a plot of accuracy vs. the value of C for varying training points shown in Figure 17. For a large number of training points (500, 960), increasing the C value increased the accuracy, whereas for a smaller number of training points (50, 100, 250), increasing the C value did not necessarily lead to a consistent increase in accuracy. In fact, it led to a decrease in accuracy for some areas. This is because as stated before, a larger value for C leads to a stricter model that overfits on the training points. For smaller training sets, this overfitting can lead to errors when applying to the test set. However, after enough increase in training points, this overfitting is alleviated as the training set becomes larger, since a model trained with a larger training set is more likely to accurately represent the entire dataset.

The effect of the smoothness hyperparameter of the Matern kernel of the Gaussian Process Regressor on the accuracy of the model was also examined. The plots in Figure 18 show that increasing the smoothness parameter leads to increasing error, and this error increases by a very large amount when the smoothness parameter is increased to 2.5, representing a very smooth function approximation. This suggests that the underlying function in the data is not represented well by a smooth function, and

Table 13: Features and Outputs of Reversibly Coilable Material with the Maximum σ_{crit} and the Maximum E_{abs} in the 3D Dataset, as predicted by the C-Support Vector Classifier

Features	Maximum σ_{crit} and Maximum E_{abs}
ratio-pitch	0.538086
ratio-d	0.042004
ratio-top-diameter	0.753125
coilable	1
sigma-crit	18.246796
energy	10.942924

Table 14: Features and σ_{crit} Value of Reversibly Coilable Material with the Maximum σ_{crit} in the 3D Dataset, as optimized with different methods

Features/Outputs	Nelson-Mead	L-BFGS-B	TNC
ratio-pitch	0.87500982	0.875	0.87500518
ratio-d	0.03850069	0.0385	0.03850064
ratio-top-diameter	0.4	0.4	0.39999715
sigma-crit	17.4150232	17.41738196	17.41592996

the actual underlying function is rather complex and non-smooth. The choice of a higher smoothness parameter thus might not be suitable for the underlying characteristics of the data.

4 Optimization

4.1 Searching for the Optimum Points Classified by the C-Support Vector Classifier

The C-Support Vector Classifier, fitted using 729 training points (75 percent of the dataset), was used to make classification predictions for all 973 points of the 3D dataset. From Part 3, it was shown a C value of 20 yielded the highest accuracy, so the SVM model had the hyperparameter $C = 20$ and the RBF kernel. Predictions were made, and the points with the maximum predicted σ_{crit} and E_{abs} were found. The maximums happen to coincide with each other, sharing the same input features. The features and inputs for the point that has the maximum σ_{crit} and E_{abs} are listed in Table 13.

4.2 Using Optimizers to Determine the Maximum Values for σ_{crit} and E_{abs}

The C-Support Vector Classifier and the Gaussian process regressor were used for the optimization problem of finding the points corresponding to the maximum values for σ_{crit} and E_{abs}. The Gaussian process regressor with the Matern kernel, with smoothness hyperparameter $\nu = 0.5$ was used, as this was the value that yielded the greatest accuracy in part 3. For the objective function to be minimized, the function would return the negative of σ_{crit} or E_{abs}, so that it sought to find the input features that would result in the greatest magnitude for σ_{crit} and E_{abs}. The function also had a nested if-else statement that made it so the objective function would return a very high value if the classifier yielded a non-coilable prediction, such that only predictions involving reversibly coilable outputs would be considered to be minimized. The three optimizers used were Nelson-Mead, L-BFGS-B, and TNC. The resulting optimized points yielding the maximum values for σ_{crit} and E_{abs} are listed in Table 11 and Table 14.

4.3 Comparison of the Solutions for the Maximum Values for σ_{crit} and E_{abs}

Overall, the maximum values for σ_{crit} and E_{abs} found within the predictions of the SVC classifier within the 3D dataset, as well as the optimized points found by the three optimizers, are all larger than the maximum values found in the original 3D dataset. Additionally, the maximum values for σ_{crit} found by the SVC predictions are very similar to the maximum values found by the three optimizers; however, the input features that yield the maximum values are different. The maximum values for E_{abs} found by the optimizers are not as similar to the maximum values for E_{abs} found by the SVC predictions, but they are still in the similar scale range. One thing to note is that the L-BFGS-B optimizer yields drastically different results when optimizing E_{abs}

Table 15: Features and E_{abs} Value of Reversibly Coilable Material with the Maximum E_{abs} in the 3D Dataset, as optimized with different methods

Features/Outputs	Nelson-Mead	L-BFGS-B	TNC
ratio-pitch	0.87501374	0.68975519	0.87504942
ratio-d	0.03850114	0.07286523	0.03849964
ratio-top-diameter	0.66724059	e	0.39996528
energy	7.29276827	55.52615479	7.29027549

when compared to the other optimizers, but seems to work similarly to the other optimizers for σ_{crit}. This is a consequence of the no free lunch theorem - optimizers work differently for different problems.

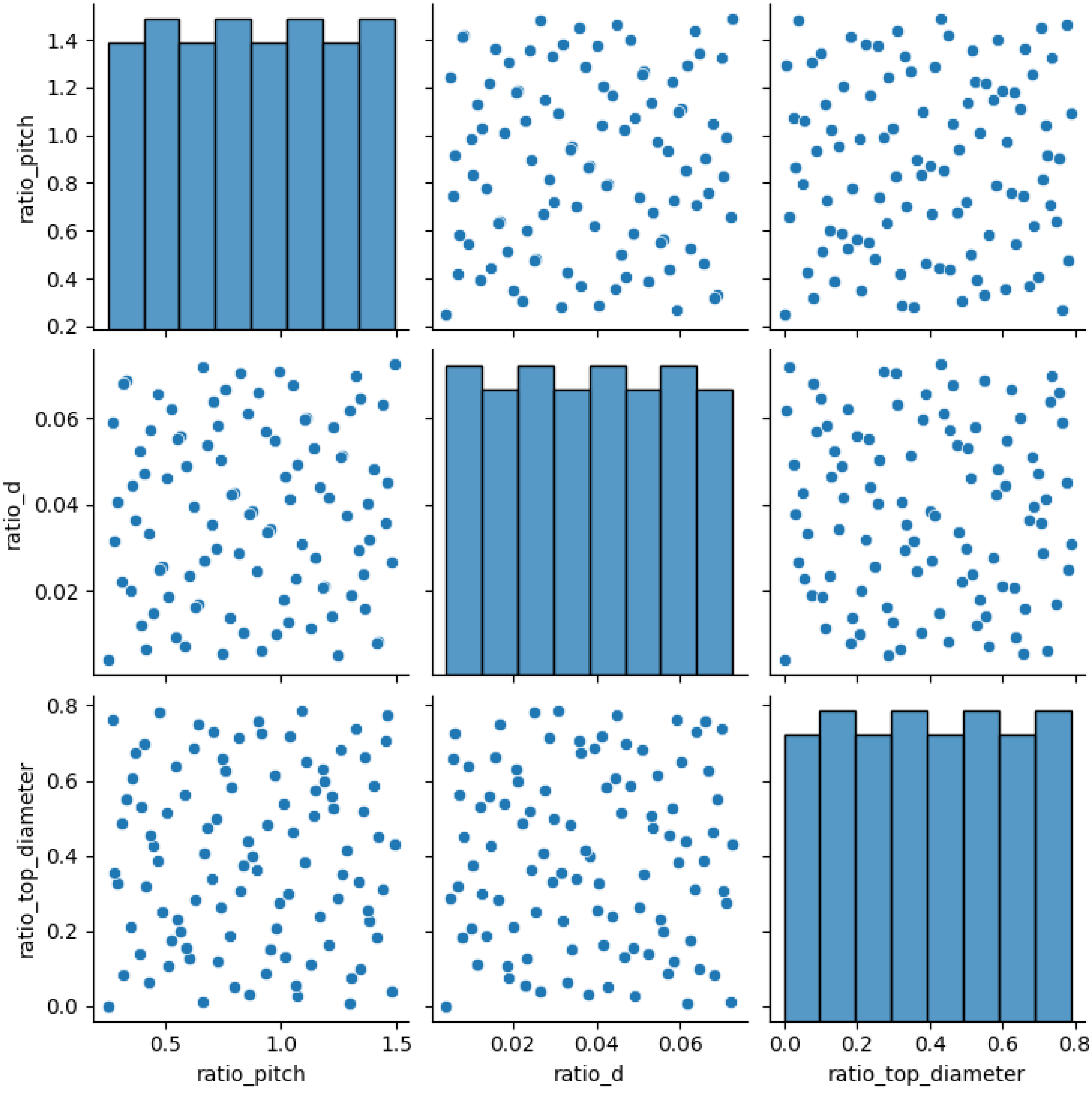


Figure 1: 2D scatter plots of combinations of the first 100 input variables

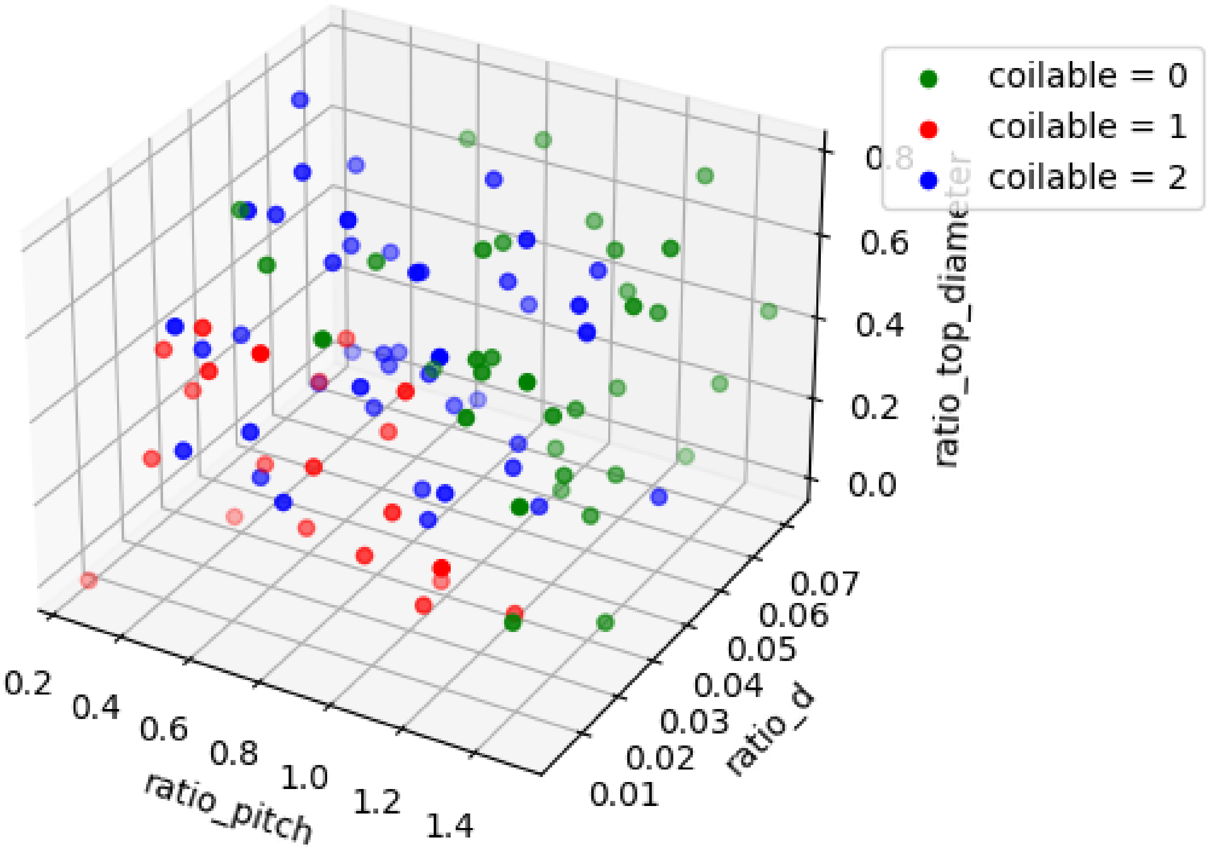


Figure 2: 3D scatter plot of the first 100 input variables

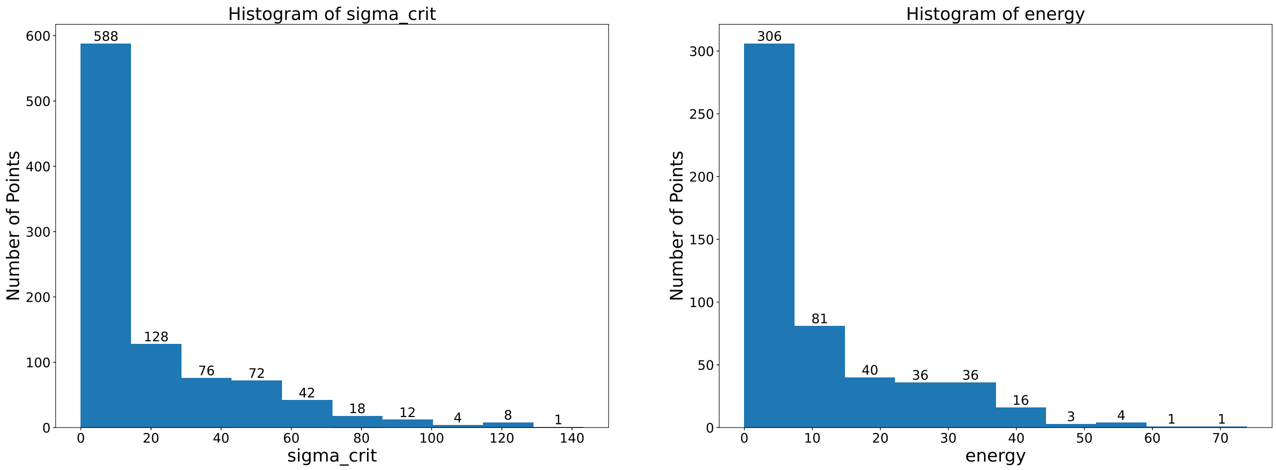


Figure 3: Histograms for the output variables in the 3D dataset

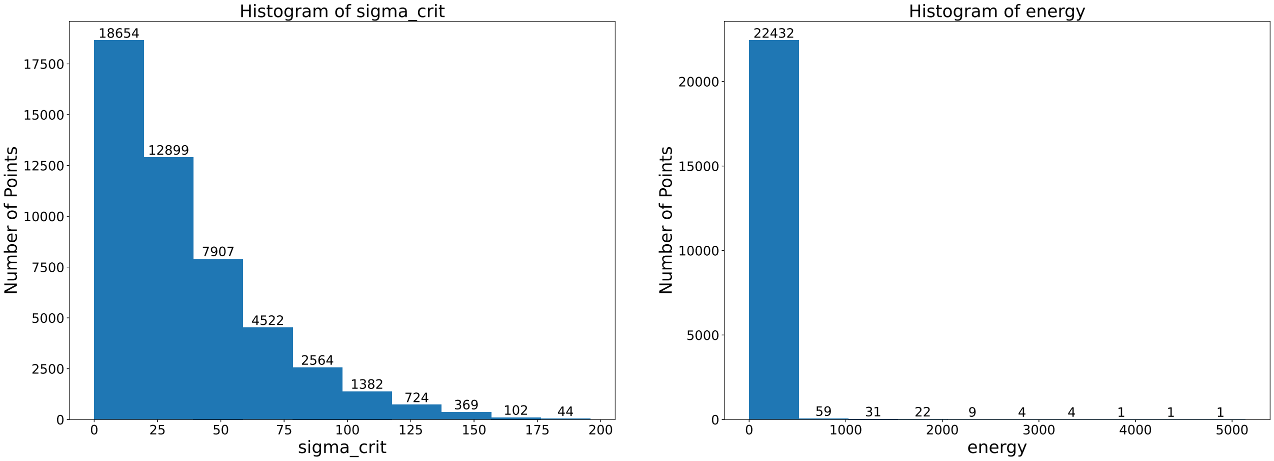


Figure 4: Histograms for the output variables in the 7D dataset

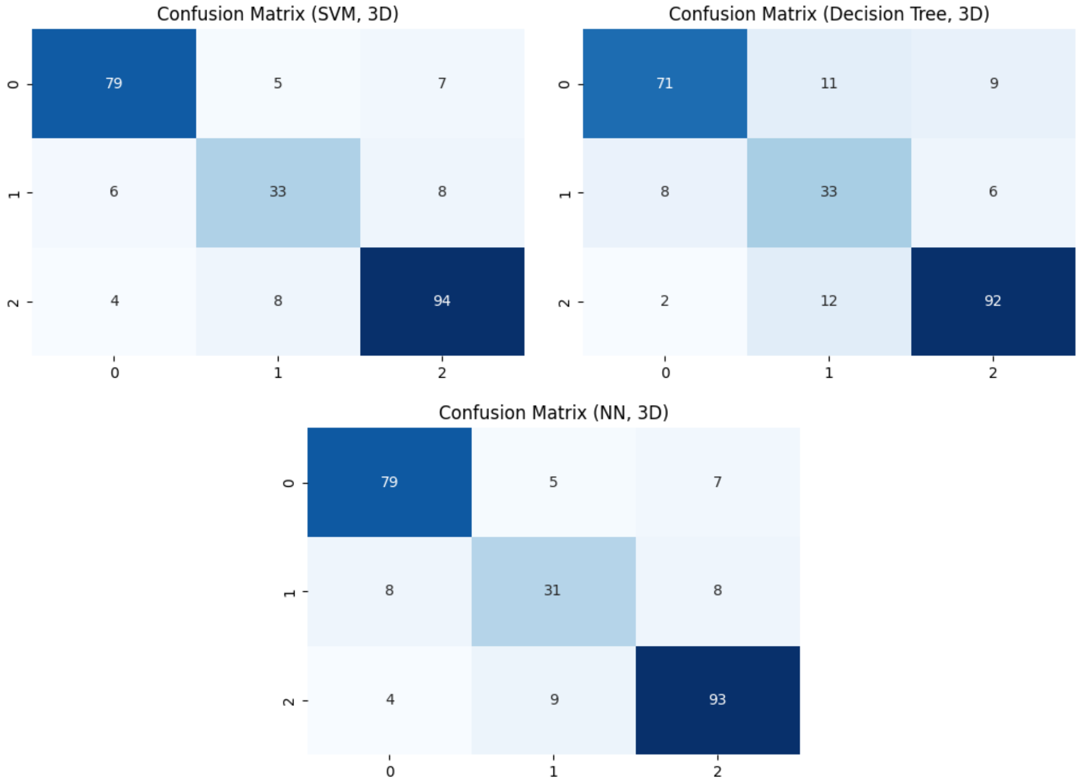


Figure 5: Confusion Matrices for the chosen classifiers on the 3D dataset

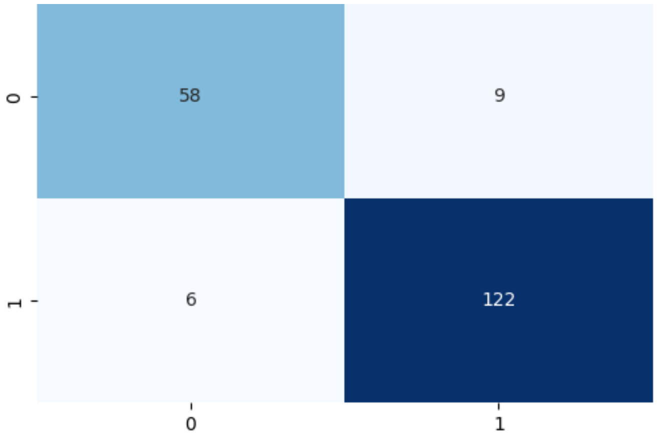


Figure 6: Confusion Matrices for all chosen classifiers on the 3D dataset (two categories)

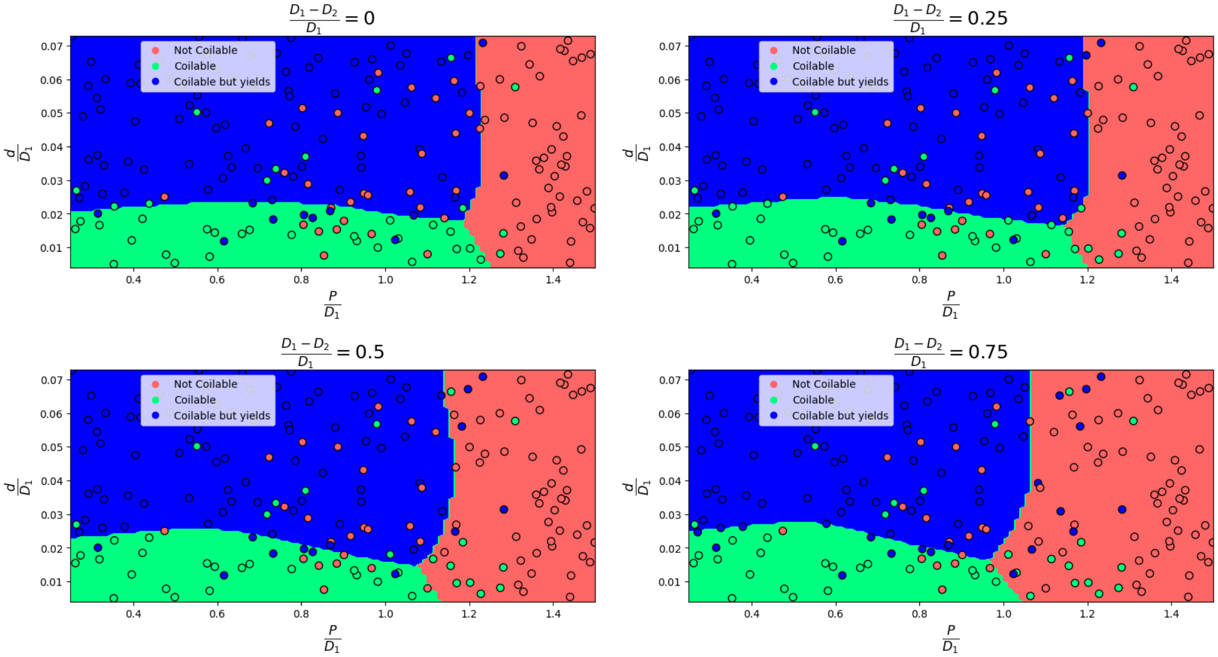


Figure 7: Classification of the 3D dataset using a SVC classifier for four different values of ratio_top_diameter.

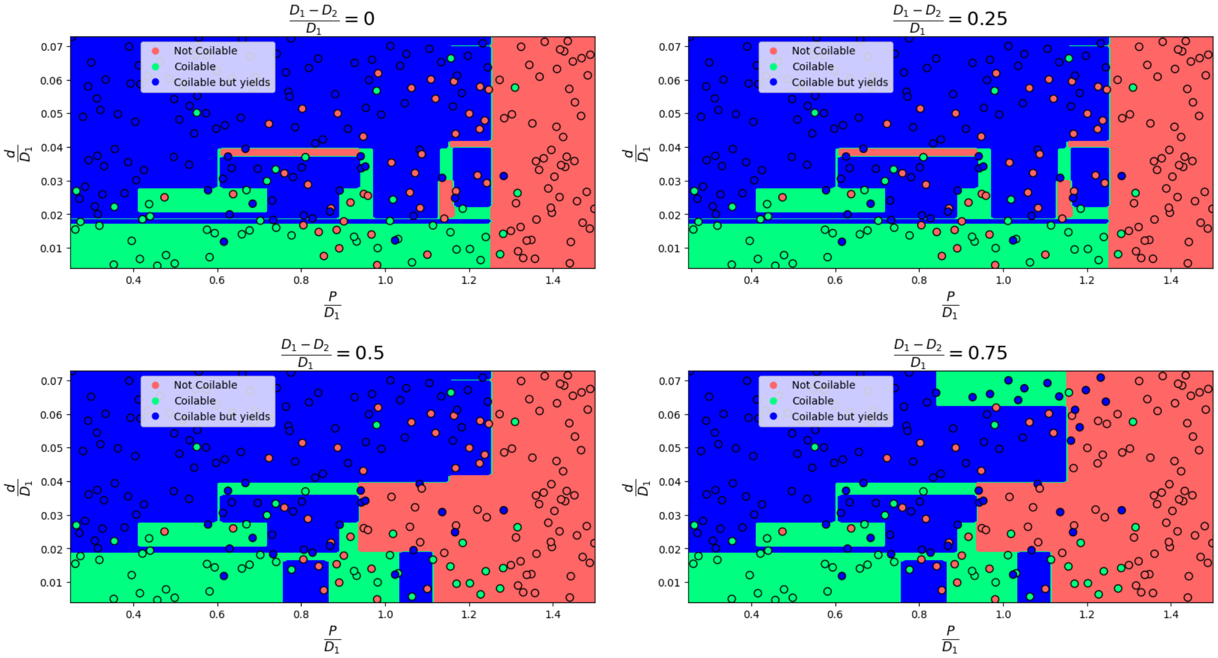


Figure 8: Classification of the 3D dataset using decision tree classifier for four different values of ratio_top_diameter.

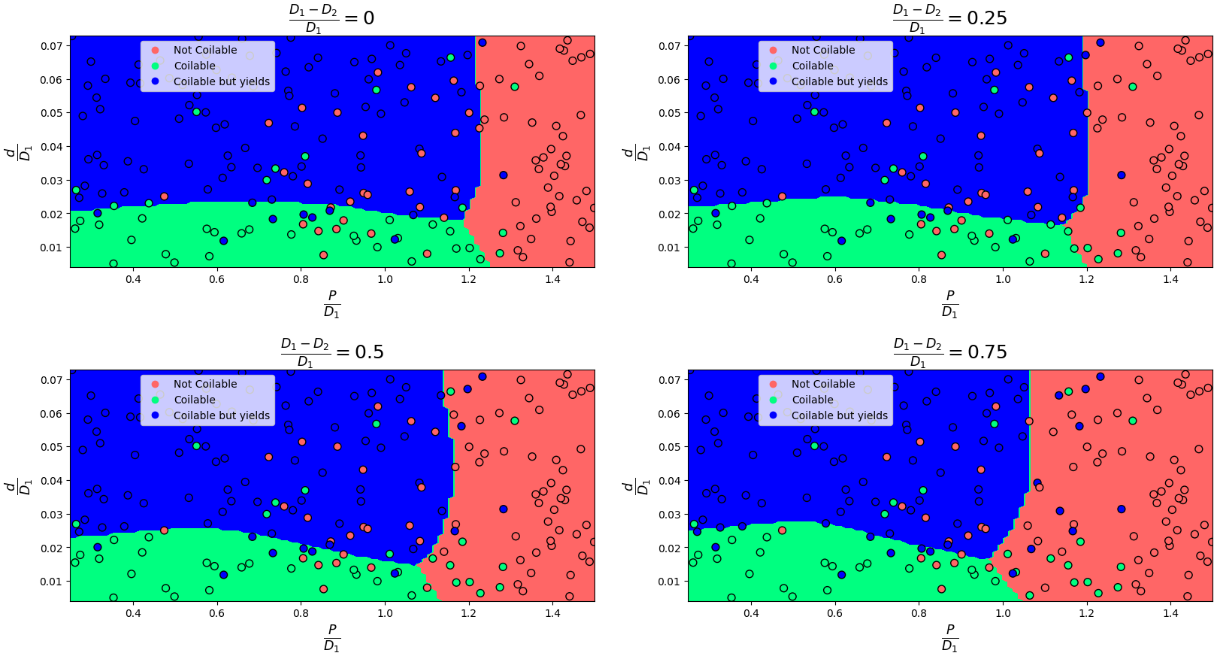
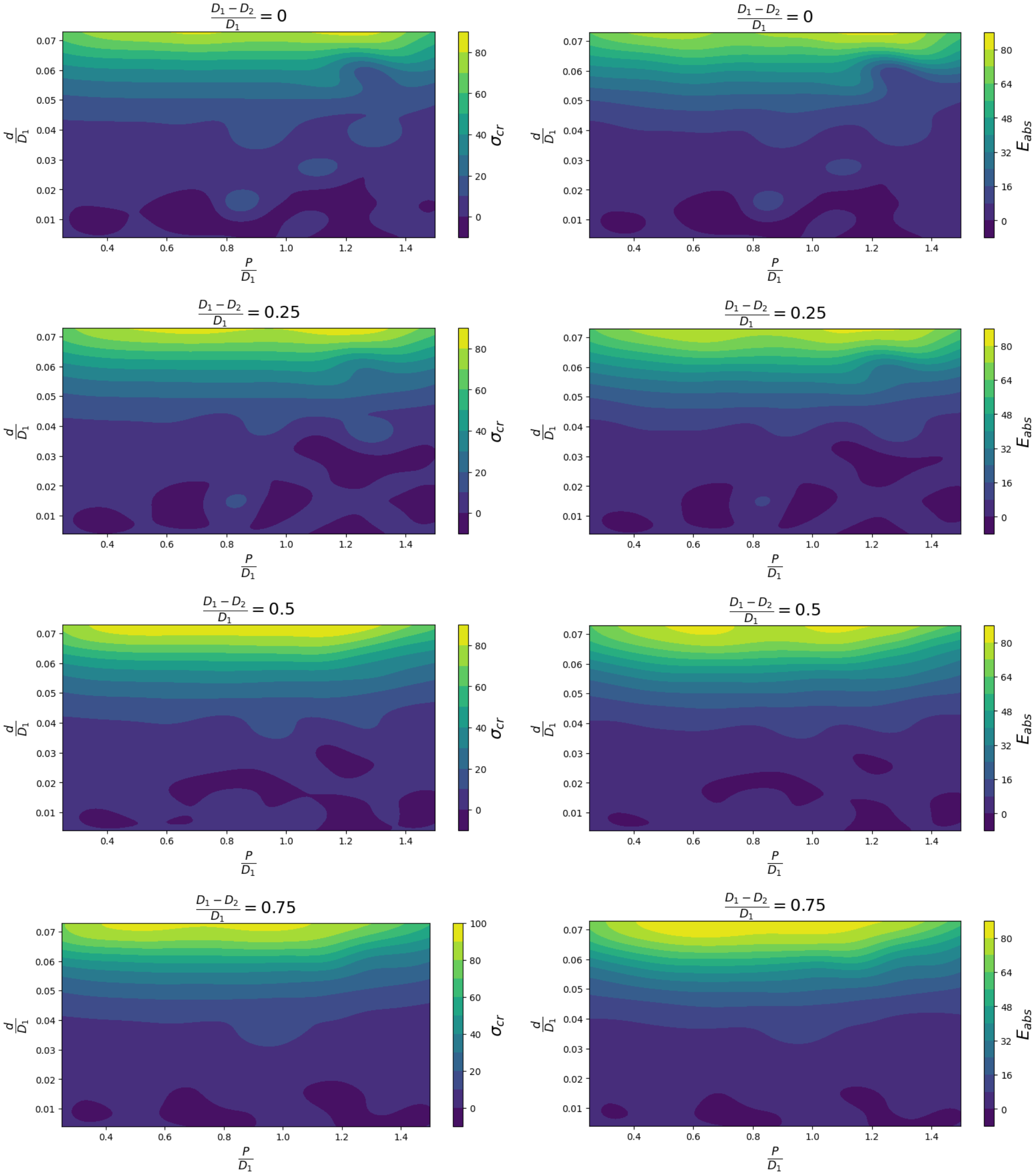


Figure 9: Classification of the 3D dataset using a NN classifier for four different values of ratio_top_diameter.



$\frac{D_1 - D_2}{D_1} = 0$

$\frac{D_1 - D_2}{D_1} = 0.25$

$\frac{D_1 - D_2}{D_1} = 0.5$

$\frac{D_1 - D_2}{D_1} = 0.75$

Figure 10: Regression analysis of the 3D dataset using Gaussian processes for four different values of the `ratio_top_diameter`.

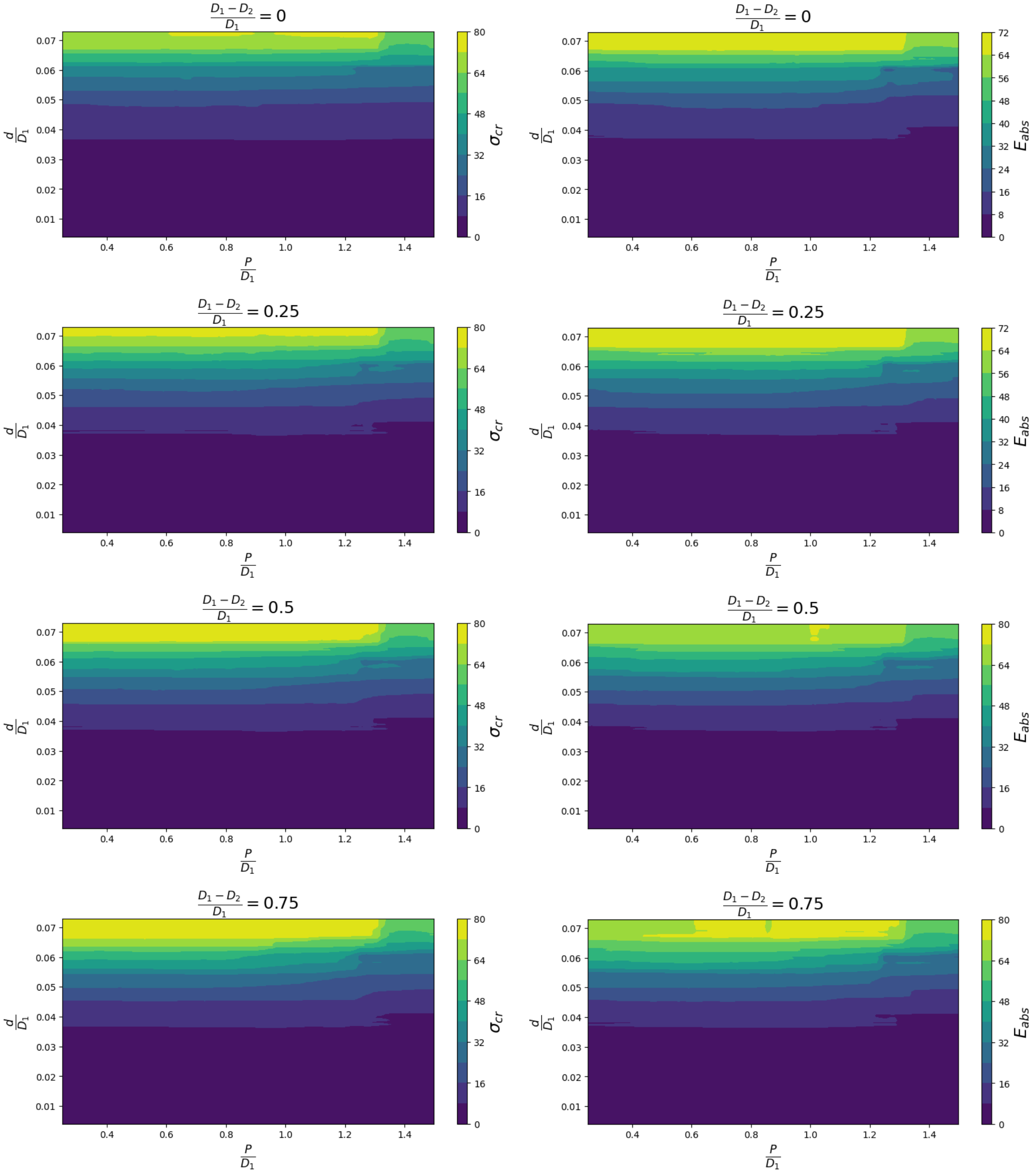


Figure 11: Regression analysis of the 3D dataset using Random Forest for four different values of the `ratio_top_diameter`.

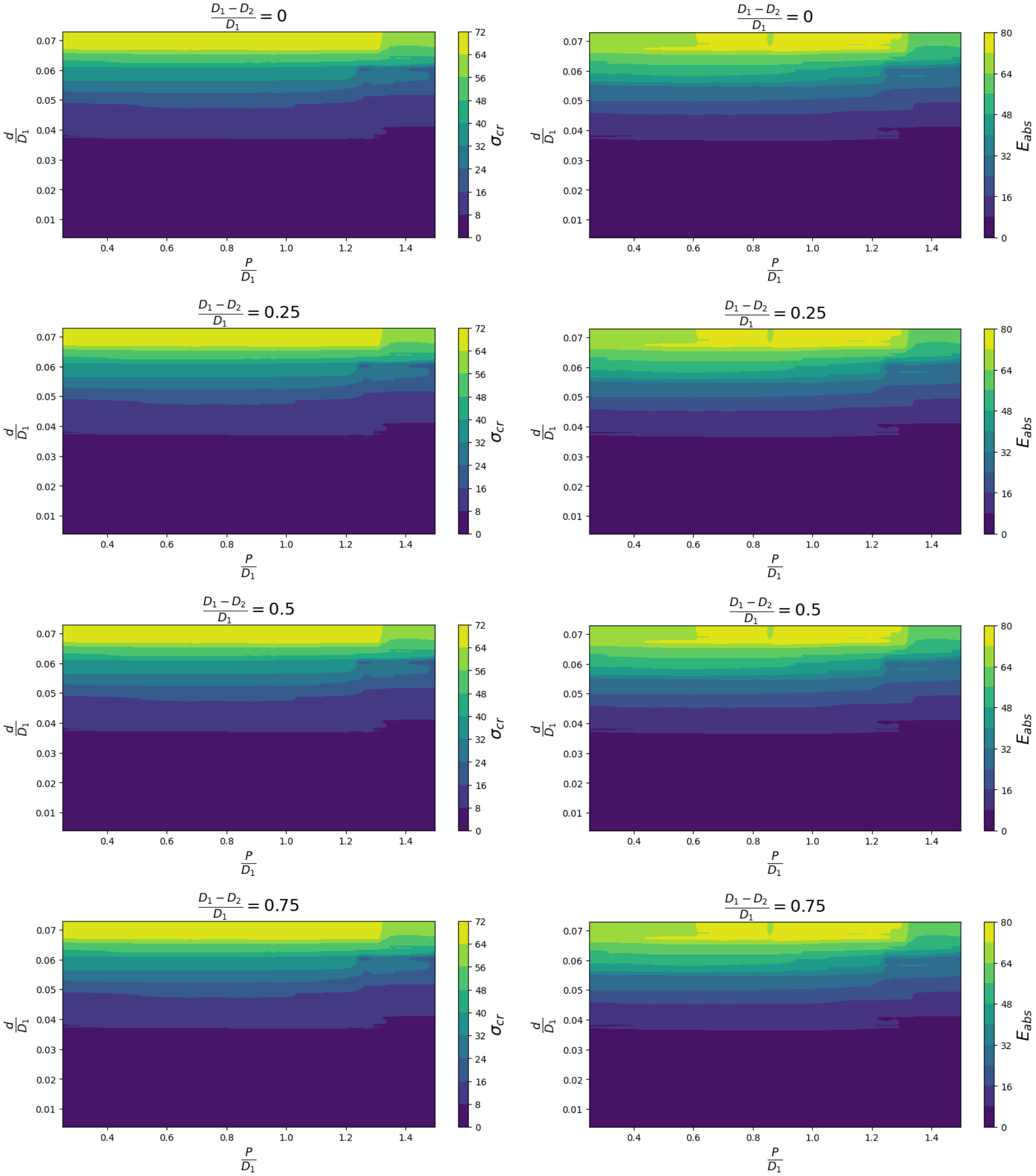
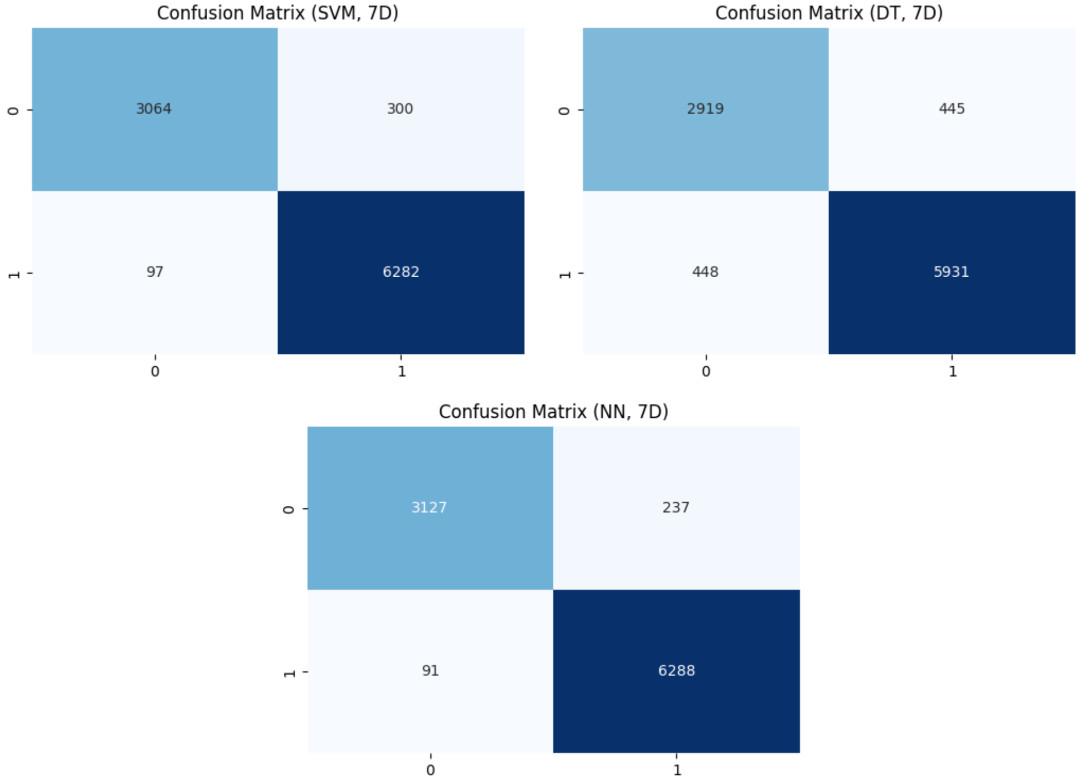


Figure 12: Regression analysis of the 3D dataset using Ridge for four different values of the `ratio_top_diameter`.



Confusion Matrix (SVM, 7D)

	0	1
0	3064	300
1	97	6282

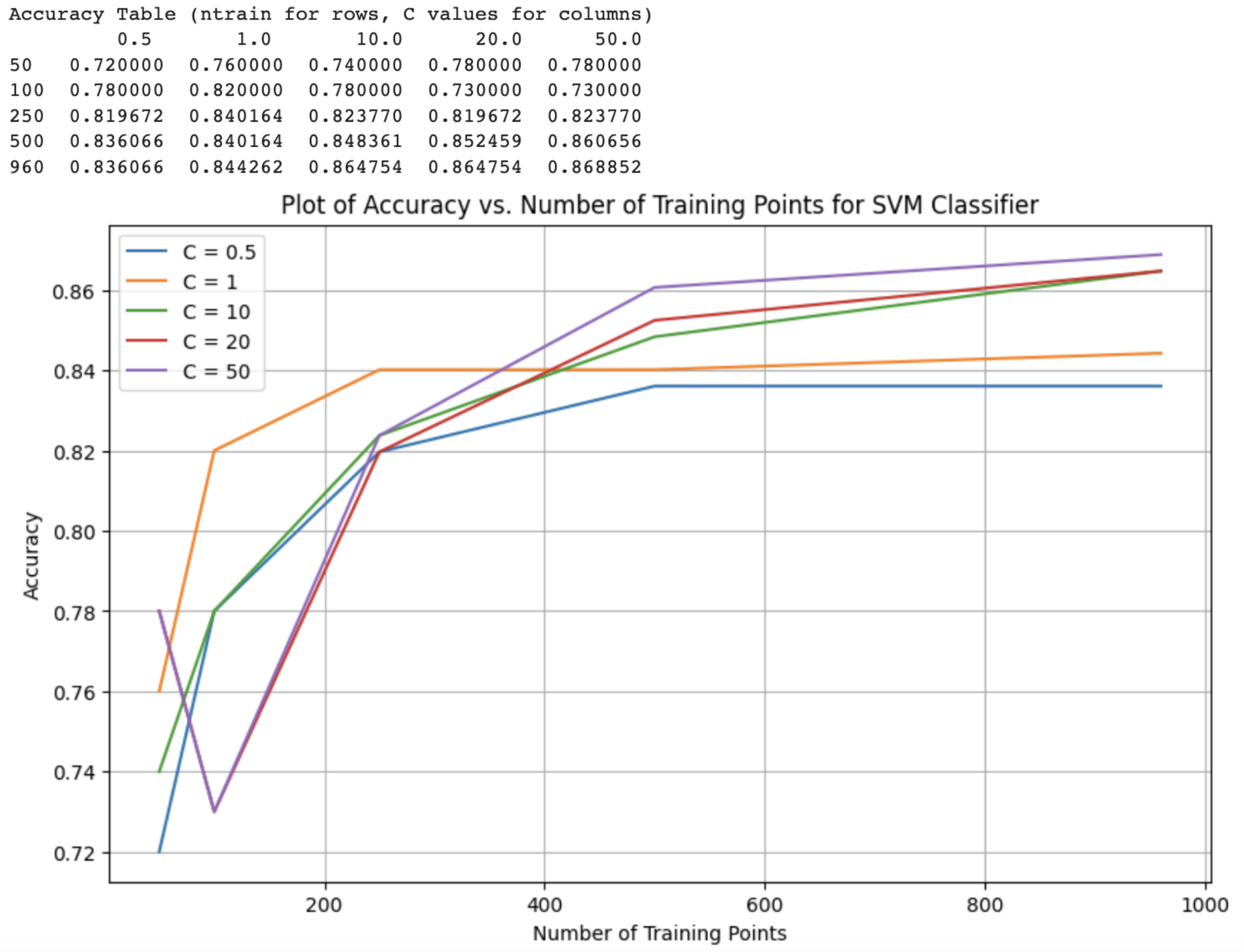
Confusion Matrix (DT, 7D)

	0	1
0	2919	445
1	448	5931

Confusion Matrix (NN, 7D)

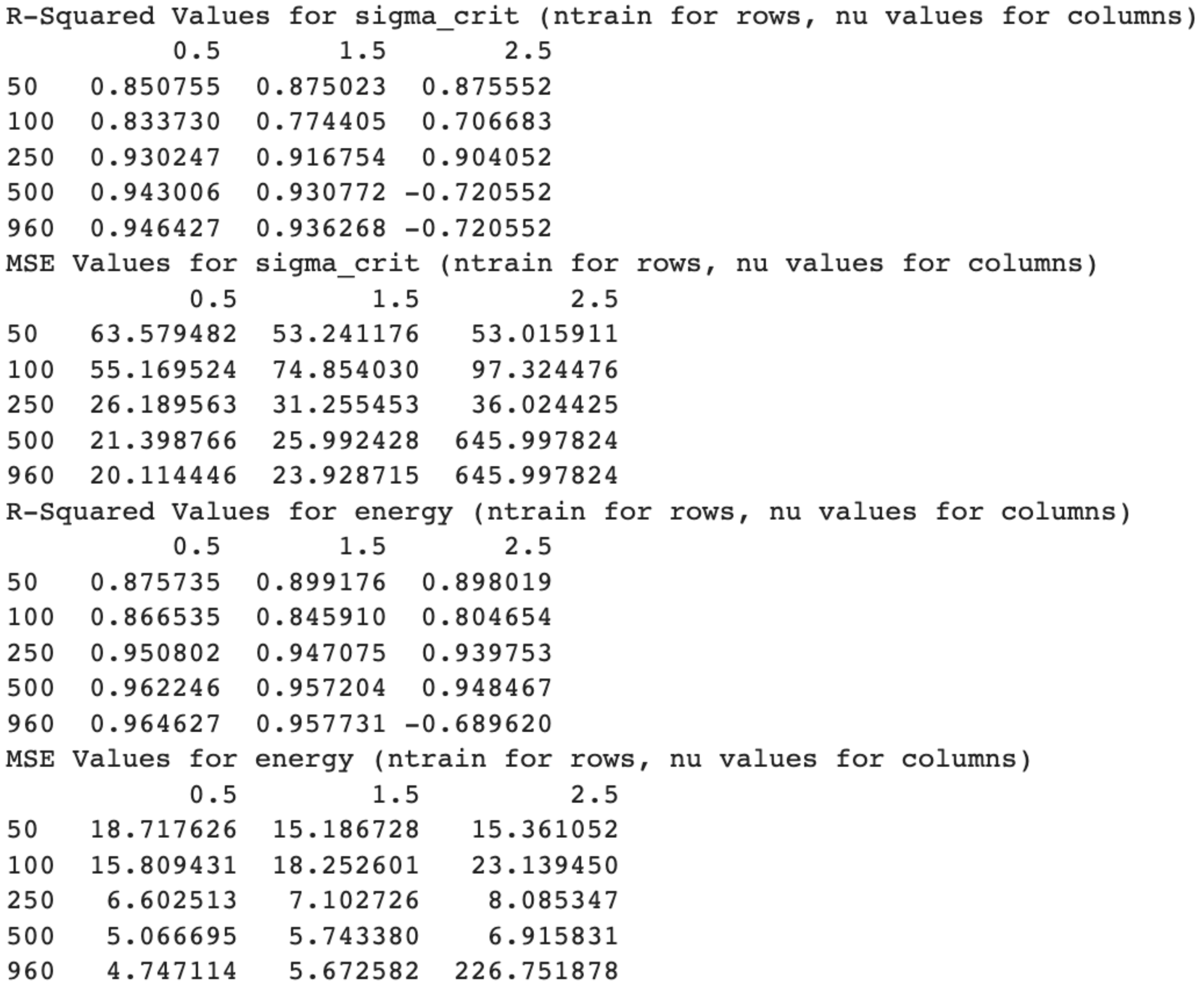
	0	1
0	3127	237
1	91	6288

Figure 13: Confusion Matrices for the chosen classifiers on the 7D dataset



```
Accuracy Table (ntrain for rows, C values for columns)
          0.5       1.0       10.0      20.0      50.0
50   0.720000  0.760000  0.740000  0.780000  0.780000
100  0.780000  0.820000  0.780000  0.730000  0.730000
250  0.819672  0.840164  0.823770  0.819672  0.823770
500  0.836066  0.840164  0.848361  0.852459  0.860656
960  0.836066  0.844262  0.864754  0.864754  0.868852
```

Figure 14: Plots of Accuracy vs. Number of Training Points used by the SVM model, with varying hyperparameter C values



```
R-Squared Values for sigma_crit (ntrain for rows, nu values for columns)
          0.5       1.5       2.5
50   0.850755  0.875023  0.875552
100  0.833730  0.774405  0.706683
250  0.930247  0.916754  0.904052
500  0.943006  0.930772 -0.720552
960  0.946427  0.936268 -0.720552
MSE Values for sigma_crit (ntrain for rows, nu values for columns)
           0.5        1.5         2.5
50   63.579482  53.241176   53.015911
100  55.169524  74.854030   97.324476
250  26.189563  31.255453   36.024425
500  21.398766  25.992428  645.997824
960  20.114446  23.928715  645.997824
R-Squared Values for energy (ntrain for rows, nu values for columns)
          0.5       1.5       2.5
50   0.875735  0.899176  0.898019
100  0.866535  0.845910  0.804654
250  0.950802  0.947075  0.939753
500  0.962246  0.957204  0.948467
960  0.964627  0.957731 -0.689620
MSE Values for energy (ntrain for rows, nu values for columns)
           0.5        1.5         2.5
50   18.717626  15.186728   15.361052
100  15.809431  18.252601   23.139450
250   6.602513   7.102726    8.085347
500   5.066695   5.743380    6.915831
960   4.747114   5.672582  226.751878
```

Figure 15: Table of R-Squared and MSE Values for vs. Number of Training Points used by the GP Model (for both sigma-crit and energy), with varying smoothness hyperparameter nu values for the Matern kernel

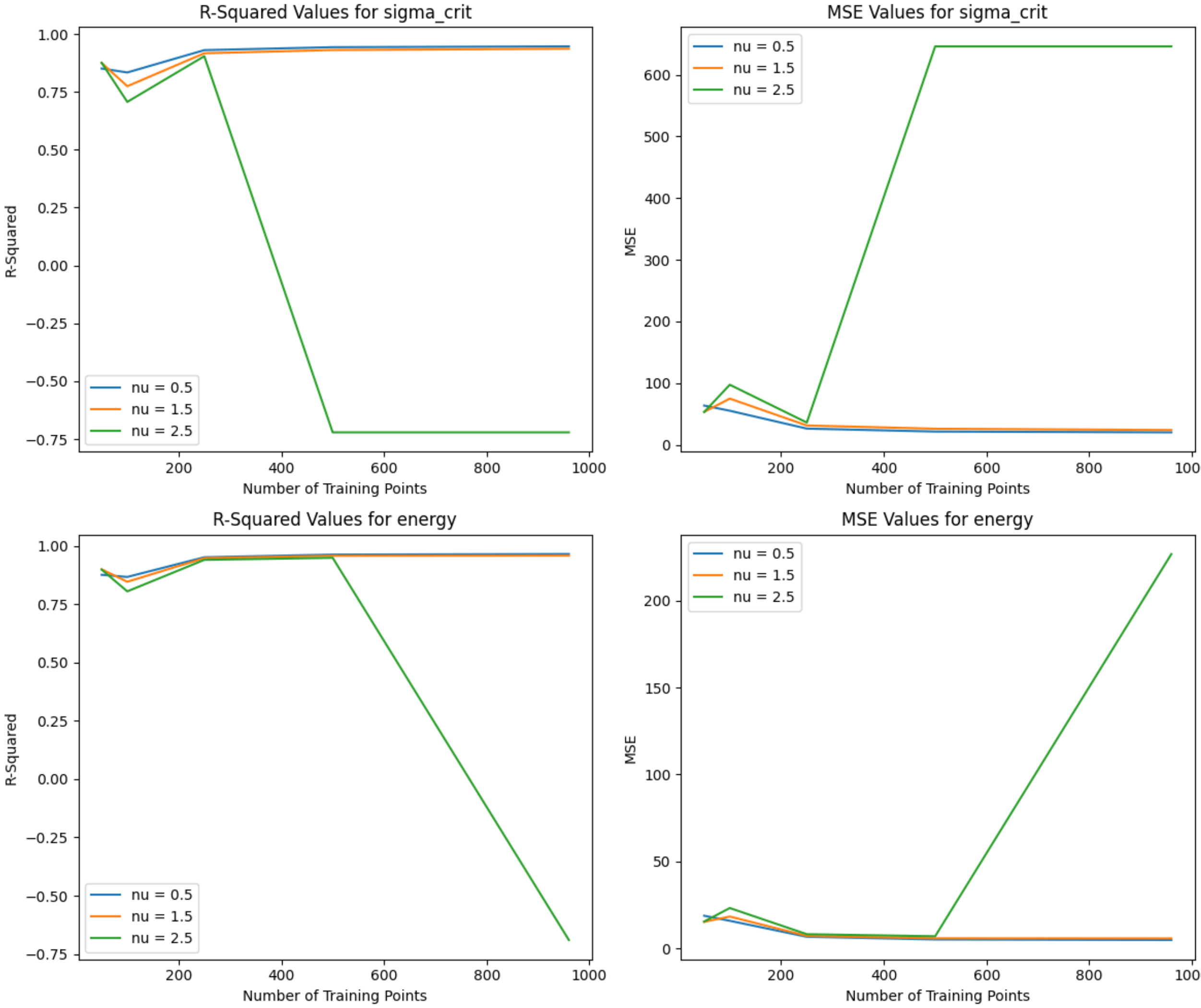


Figure 16: Plots of R-Squared and MSE Values for vs. Number of Training Points used by the GP Model (for both sigma-crit and energy), with varying smoothness hyperparameter nu values for the Matern kernel

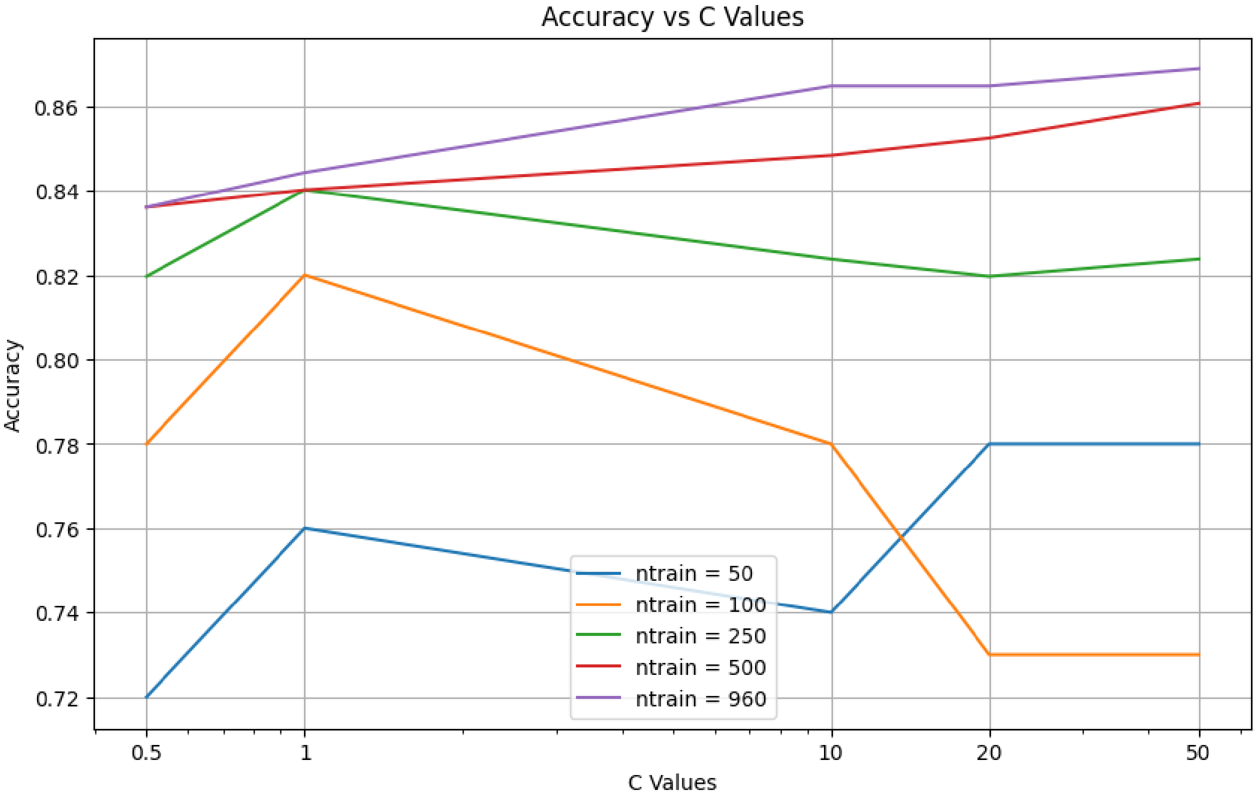


Figure 17: Plot of Accuracy vs. C Values used by the SVM Model, with varying number of training points

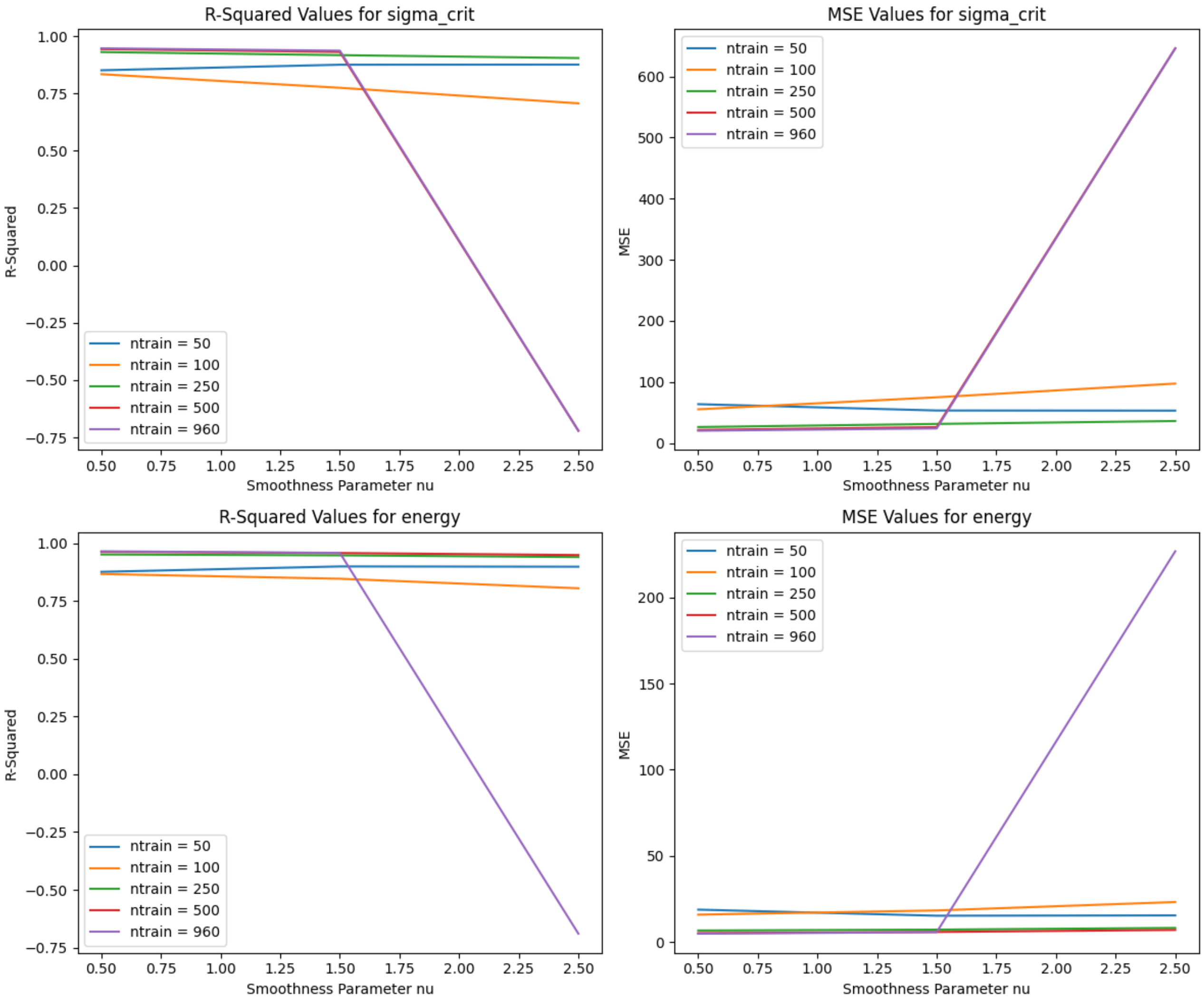


Figure 18: Plots of R-Squared and MSE Values for vs. Smoothness Parameter used by the C-Support Vector Classifier (for both sigma-crit and energy), with number of training points