Technical University of Denmark

# Gas Turbine Emissions: A Classification and Regression Task

02450 Introduction to Machine Learning and Data Mining

Jonas Wiendl, s243543
Linnea Hockauf, s204547
Rami Ishac Hanna, s242507

November 13, 2024

# Abstract

This report builds on the initial data analysis and exploration, applying machine learning techniques to predict emissions (NOx and CO) from gas turbines. Using a dataset with features related to environmental conditions, turbine operating parameters, and mechanical pressure control, we focused on selecting suitable models and investigating the influence of model parameters on performance. For NOx emissions, Ridge regression and an artificial neural network (ANN) were implemented, with a baseline model serving as a benchmark. Results indicated that the ANN model achieved the lowest mean squared error (MSE), outperforming both Ridge regression and the baseline model, capturing complex relationships in the data effectively.

For CO emissions, we used a binary classification approach to categorize values as high or low, comparing three models: a baseline majority classifier, logistic regression with L2 regularization, and k-nearest neighbors (KNN). The KNN model performed best, effectively managing class imbalance and capturing local data patterns. Statistical tests confirmed the superiority of the ANN and KNN models over simpler baselines.

These results show that advanced machine learning models, especially those able to handle non-linear relationships, are effective for predicting gas turbine emissions. Regularization and cross-validation helped create stable models that work well on new data. Although ANN and KNN models provided strong accuracy, linear models like Ridge regression and logistic regression were useful for interpreting relationships between features in a practical context.

# Contribution

*Table 1: Contribution*

| | Jonas Wiendl | Linnea Hockauf | Rami I. Hanna |
|---|---|---|---|
| Data Understanding and Preparation | 34 | 33 | 33 |
| Regression | 50 | 25 | 25 |
| Classification | 25 | 40 | 35 |
| Discussion | 25 | 50 | 25 |
| Exam Problems | 33 | 33 | 33 |

# Contents

# 1 Data Understanding and Preparation

This report builds on the foundation established in the first analysis, focusing on sensor data from gas turbines with the goal of predicting emissions. The dataset contains 9 features and 2 targest variables: Nitrogen Oxides (NOx) for regression and Carbon Monoxide (CO) for classification.

Before modeling, the data was standardized using the StandardScaler from sklearn, ensuring that each feature had a mean of 0 and a standard deviation of 1. By using regularization, we ensure that all features contribute equally to the model and thereby improve the performance and stability of the models.

# 2 Regression

In this section we discuss Ridge Regression, Linear Regression with L2 normalization to predict Nitrous Oxide (NOx). To do so, the model assumes a linear relationship:

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_K x_K$$

Ridge Regression is effective at preventing overfitting, by adding the regularization term lambda, $\lambda$, to our loss function:

$$E_\lambda = \left\| \hat{\mathbf{y}} - \hat{\mathbf{X}}\mathbf{w} \right\|^2 + \lambda \left\| \mathbf{w} \right\|^2$$

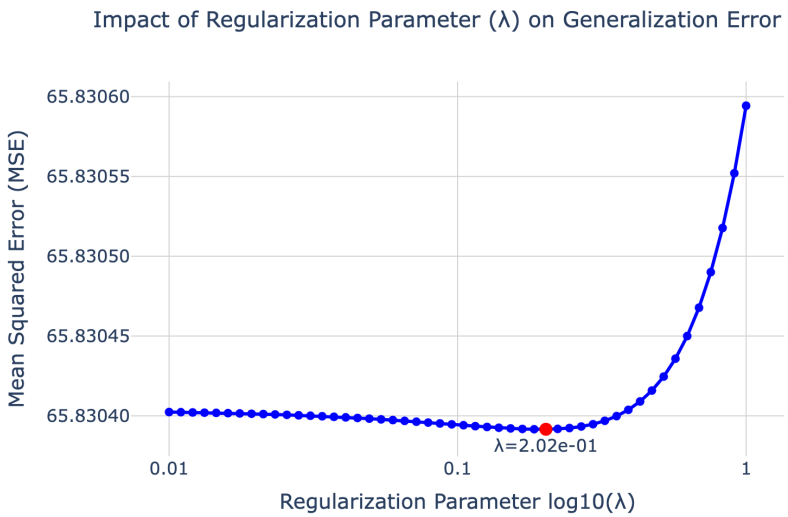


*Figure 1: Impact of L2 Regularization on Generalization*

To evaluate model performance, we tested a wide range of $\lambda$ values from $10^{-4}$ to $10^{4}$. Additionally, a $K = 10$ fold cross-validation was used to ensure a reliable estimation of the generalization error. As shown in Figure 1, the smallest generalization error is achieved at $\lambda = 2.02 \times 10^{-1}$, indicating a balanced trade-off between model complexity and regularization, effectively preventing both over- and underfitting.

In Table 2, the coefficients of the final Ridge Regression model are presented with the ambient temperature exhibiting a strong negative coefficient, suggesting that emissions increase as the temperature decreases. Conversely, the turbine inlet temperature shows a significant positive coefficient, indicating that emissions rise with higher inlet temperatures, which aligns with expectations given the operational dynamics of gas turbines.

*Table 2: Coefficients for Each Feature*

| **Feature** | **Coefficient** |
|---|---|
| Ambient temperature | -13.12 |
| Ambient pressure | -1.52 |
| Ambient humidity | -3.22 |
| Air filter difference pressure | 0.54 |
| Gas turbine exhaust pressure | -0.47 |
| Turbine inlet temperature | 24.70 |
| Turbine after temperature | -10.45 |
| Turbine energy yield | -30.42 |
| Compressor discharge pressure | -1.93 |

## 2.1 Comparative Analysis of Regression Models

This section presents a comparison between Ridge Regression, a single-layer artificial neural network (ANN), and a baseline model. In order to ensure a fair and consistent comparison of their performance, the same two level cross-validation ($K_1 = K_2 = 10$) setup is used.

### 2.1.1 Baseline Model

The baseline model provides a simple benchmark by predicting the mean value of the target variable from the training data for every sample in the test set. Specifically, for each outer fold, the mean of the training target values, $\bar{y}_{\text{train}}$, is calculated as followed and used as the prediction for all test samples.

$$\bar{y}_{\text{train}} = \frac{1}{N} \sum_{i=1}^{N} y_i$$

### 2.1.2 Linear Regression

Building on the concepts introduced at the beginning of this chapter, we apply Ridge Regression to model the relationship between the features and the target variable. To find the optimal regularization parameter $\lambda$, we use the inner loop of our two-level cross-validation framework. Based on the previous results the values for $\lambda$ are tested in the range:

$$\lambda_{\text{values}} = \{0.1, 0.144, 0.189, 0.233, 0.278, 0.322, 0.367, 0.411, 0.456, 0.5\}$$

### 2.1.3 Artificial Neural Network

To further enhance our regression analysis, we employed a Multi-Layer Perceptron (MLP) model from the `scikit-learn` library. Each neuron in the hidden layer is a model parameter we can expirement with.

An exploratory analysis of neuron count was performed. As shown in Figure 2, the model error decreases as the number of neurons increases, with the performance stabilizing for $h > 750$. No signs of overfitting were demonstrated so increasing the number of neurons could increase performance but would also raise the complexity of the model. (The impact of adding more layers to further increase complexity was conducted and shown in Appendix A, just outside the scope of the project).

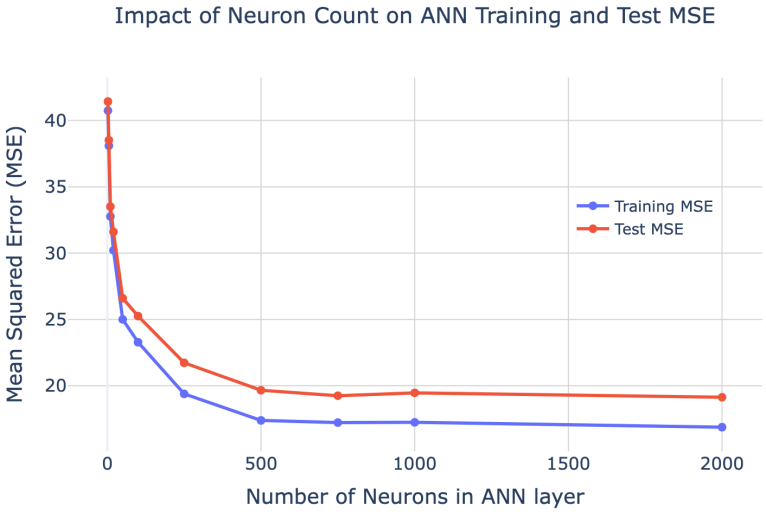


*Figure 2: Impact of Number of Neurons in ANN layer on MSE*

The following values were selected for fine-tuning the model:

$$h_{\text{values}} = \{1, 175, 250, 325, 400, 475, 550, 625, 700, 750\}$$

### 2.1.4 Model Comparison and Statistical Evaluation

To evaluate the performance of our regression models, we compared the Mean Squared Error (MSE) per observation across all outer folds for the Artificial Neural Network (ANN), Ridge Regression, and the Baseline Model. Additionaly, to gain a better understanding of the percent error of the models, the nRMSE can be introduced as:

$$\text{nRMSE} = \frac{\sqrt{\text{MSE}}}{\text{mean}(y)}$$

From the results presented in Figure 3 and Table 4, it is evident that the ANN model consistently achieves the lowest MSE across all outer folds, with an average MSE of 19.44 (nRMSE of 4.41%), indicating superior predictive performance. Ridge Regression performs moderately, with MSE values significantly lower than the Baseline Model but higher than the ANN model, averaging at 65.83 (12.43%). The Baseline Model, as expected, has the highest MSE, averaging 136.39 (17.89%).

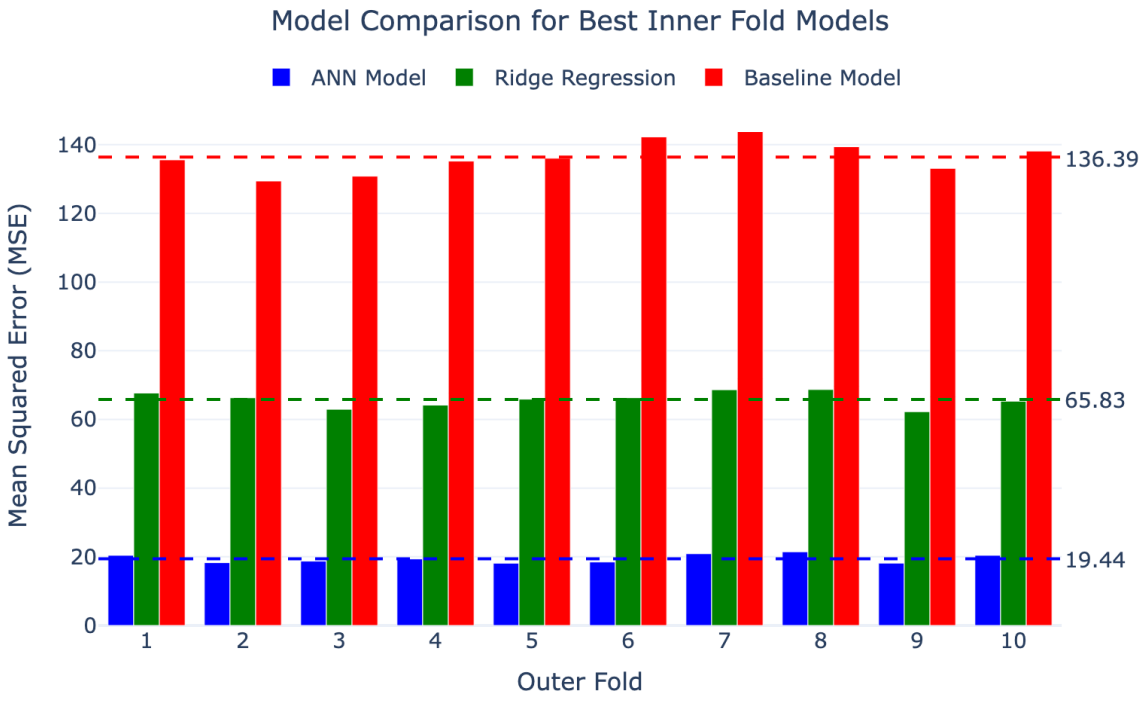


*Figure 3: Model Comparison for Best Inner Fold Models*

In contrast to Figure 1, Table 4 highlights varying $\lambda$ values across models leading to differences in the regularization parameter. Figure 1, shows the generalization error having minimal sensitivity to changes in $\lambda$, explaining the small differences in model performance.

| **Outer fold** $i$ | **ANN** | | **LR** | | **Baseline** |
|---|---|---|---|---|---|
| | $h_i^*$ | $E_i^{\text{test}}$ | $\lambda_i^*$ | $E_i^{\text{test}}$ | $E_i^{\text{test}}$ |
| 1 | 475 | 20.43 | 0.32 | 67.69 | 135.60 |
| 2 | 550 | 18.29 | 0.37 | 66.30 | 129.43 |
| 3 | 550 | 18.73 | 0.10 | 62.96 | 130.84 |
| 4 | 625 | 19.38 | 0.10 | 64.18 | 135.22 |
| 5 | 750 | 18.14 | 0.19 | 65.90 | 136.06 |
| 6 | 625 | 18.48 | 0.10 | 66.33 | 142.28 |
| 7 | 625 | 20.90 | 0.19 | 68.65 | 143.80 |
| 8 | 700 | 21.47 | 0.50 | 68.70 | 139.41 |
| 9 | 750 | 18.17 | 0.14 | 62.27 | 133.13 |
| 10 | 325 | 20.43 | 0.23 | 65.32 | 138.18 |

*Table 3: Comparison of ANN, Linear Regression, and Baseline MSE across Outer Folds*

To determine whether there is a significant performance difference among the models, a series of pairwise statistical comparisons was conducted. As the data source is limited, setup I was chosen and paired t-tests as described in chapter 11 of the lectures notes were used. The results of the tests are summarized in Table 4.

| **Comparison** | **Mean Difference** ($\bar{z}$) | **Variance** ($\hat{\sigma}^2$) | **95% CI** | **p-value** |
|---|---|---|---|---|
| ANN vs. LR | -46.39 | 0.2736 | [-47.57, -45.21] | 1.49e-14 |
| ANN vs. Baseline | -116.95 | 1.6712 | [-119.88, -114.03] | 1.24e-14 |
| LR vs. Baseline | -70.56 | 1.3933 | [-73.23, -67.89] | 5.17e-13 |

*Table 4: Statistical Comparison of Model Performances*

The comparison between the ANN, Ridge Regression, and baseline models reveals significant performance differences. The ANN outperforms Ridge Regression by a mean difference of 46.39, ($p = 1.49 \times 10^{-14}$), with a confidence interval not including zero. The second comparison, between the ANN and Baseline model shows an even larger mean difference of $-116.95$, ($p = 1.24 \times 10^{-14}$), with similar significance in the results. Ridge Regression also outperforms the Baseline with a difference of 70.56, (p -5.17 $\times$ $10^{-13}$), not reaching performance level of the ANN. Each confidence interval excludes zero, emphasizing the differences in model performance are statistically significant. These findings lead to the conclusion that both models are better than the Baseline model, with the ANN model being best.

# 3 Classification

The binary classification task of CO categorizes values as *high* or *low*, with a threshold of CO = 2.5 mg/m$^3$, resulting in a 60:40 split favoring *low* values, with 26,009 v*low* and 10,724 *high* values. The classification performance is evaluated by comparing three models: Baseline model using a Majority Classifier (MC) as a simple benchmark and Logistic Regression (LR), and a K-Nearest Neighbors Classifier (KNN) to handle class imbalance and capture local data patterns. To adjust for potential overfitting, the Logistic Regression includes a L2 regularization with $\lambda$ optimized from $10^{-2}$ to $10^2$, while the KNN model is tuned for neighbors ($k$) from 1 to 20. The performance of all 3 models were evaluated using a two-level 10-fold cross validation. Only the results from the outer fold corresponding to the best parameter from the inner fold are reported. The performance metrics, including the error rate ($E_i^{\text{test}}$) are shown in **Table 5** for each other fold. The same dataset splits were used across all outer folds for consistency, with KNN's number of neighbors $k$ represented by $k_i^*$.

*Table 5: Comparison of KNN, Logistic Regression, and Baseline Error Rates across Outer Folds*

| **Outer fold** | **KNN** | | **LR** | | **Baseline** |
|---|---|---|---|---|---|
| $i$ | $k_i^*$ | $E_i^{\text{test}}$ | $\lambda_i^*$ | $E_i^{\text{test}}$ | $E_i^{\text{test}}$ |
| 1 | 9 | 0.065 | 10 | 0.096 | 0.286 |
| 2 | 9 | 0.074 | 0.01 | 0.106 | 0.297 |
| 3 | 3 | 0.063 | 1.01 | 0.096 | 0.298 |
| 4 | 3 | 0.065 | 0.1 | 0.097 | 0.285 |
| 5 | 5 | 0.065 | 10 | 0.088 | 0.288 |
| 6 | 3 | 0.068 | 0.01 | 0.098 | 0.288 |
| 7 | 5 | 0.069 | 0.01 | 0.096 | 0.304 |
| 8 | 3 | 0.068 | 0.1 | 0.096 | 0.296 |
| 9 | 7 | 0.064 | 10 | 0.091 | 0.291 |
| 10 | 3 | 0.063 | 10 | 0.091 | 0.288 |

From **Table 5** it is visible that the baseline model performs consistently worse than both the KNN and logistic regression model in terms of error rate. Moreover, the KNN model exhibits the lowest error rate among the 3 models. Further assesment of the 3 models was conducted using McNemar's test for statistical comparison. The resulting contigency tables for each pairwise comparison show the models' performance in terms of true positive, false positives, true negatives, and false negatives.

$$MC\ v\ LR = \begin{bmatrix} 24833 & 8391 \\ 1176 & 2333 \end{bmatrix} \quad MC\ v\ KNN = \begin{bmatrix} 25095 & 9204 \\ 914 & 1520 \end{bmatrix} \quad LR\ v\ KNN = \begin{bmatrix} 32392 & 832 \\ 1907 & 1602 \end{bmatrix}$$

From these tables, it is immediately apparent that both the Logistic Regression and KNN models outperform the baseline model (Majority Classifier). The estimated performance difference ($\theta$), its confidence interval, and p-values derived from McNemar's test are summarized in **Table 6**.

*Table 6: McNemar Test Results for Model Comparisons*

| **Comparison** | **Performance Difference ($\theta$)** | **95% CI** | **p-value** |
|---|---|---|---|
| MC vs. KNN | 0.8193 | [0.7514, 0.8873] | 0.0 |
| MC vs. LR | 0.7542 | [0.6931, 0.8152] | 0.0 |
| KNN vs. LR | -0.3925 | [-0.4739, -0.3110] | $1.38 \times 10^{-93}$ |

The McNemar's test results in **Table 6** indicate statistically significant performance difference ($\theta$) and confidence intervals, confirming that both KNN and Logistic Regression outperform the Majority Classifier (baseline model), with positive $\theta$ values and confidence intervals excluding zero. In contrast, the negative $\theta$ value between KNN and Logistic Regression, also significant, identifies KNN as the best-performing model for this dataset. Building on these results, investigation of **Table 5** reveals that $\lambda = 10$ exhibits the lowest error rate for Logistic Regression, making it the optimal choice of hyperparameter. The weighted features are displayed in **Table 7** corresponding to a $\lambda$ value of 10.

*Table 7: Coefficients for Each Feature for $\lambda = 10$*

| **Feature** | **Coefficient** |
|---|---|
| Ambient temperature | 1.25 |
| Ambient pressure | -1.21 |
| Ambient humidity | 0.02 |
| Air filter difference pressure | -0.35 |
| Gas turbine exhaust pressure | 0.17 |
| Turbine inlet temperature | 0.73 |
| Turbine after temperature | 2.17 |
| Turbine energy yield | -2.63 |
| Compressor discharge pressure | -4.23 |

The feature coefficients indicate a strong negative impact on CO levels from compressor discharge pressure, turbine energy yield, and ambient pressure, while turbine after temperature and ambient temperature show a positive impact. Notably, while turbine energy yield negatively affects both CO and NOx, ambient temperature and turbine after temperature positively impact CO but negatively affect NOx, highlighting distinct mechanisms behind each emission type.

# 4 Discussion

In this report, regression and classification models significantly outperformed baseline models, demonstrating the effectiveness of advanced modeling techniques for gas turbine emissions data. Using K-Fold Cross Validation, including two-layered setups, enabled effective hyperparameter tuning and reduced overfitting. The dataset's cleanliness likely contributed to minimal variation across $K$ and $\lambda$ values, suggesting that complex preprocessing was not required. For the regression, the optimal $\lambda = 0.202$ effectively balanced model complexity and generalization. While the ANN model achieved the lowest MSE, Ridge Regression provided key insights into feature relationships. It showed that ambient and turbine inlet temperatures are linked to NOx levels, while other factors like turbine energy yield and ambient pressure affect CO. Notably, some features impact CO and NOx in opposing ways, highlighting distinct mechanisms behind each emission. For classification, KNN performed best with the lowest error rate, likely due to its ability to capture local data patterns, while Logistic Regression effectively handled the CO class imbalance, outperforming the baseline.

Overall, regularization and cross-validations proved essential for achieving stable, generalizable models. The performance of ANN and KNN suggests that nonlinear models may be better suited for this dataset. However, Ridge Regression and Logistic regression offered valuable insight into feature interpretability, especially useful for real-world applications and understanding of emission influences.

## 4.1 Previous Works

The dataset in the report was analyzed by Kaya et al. in 2019 [1]. They used Extreme Learning Machines (ELMs) and achieved benchmark results for regression tasks that outperformed simpler models. Their analysis reported similar findings, in particular the effect of features like turbine energy yield and compressor discharge pressure on the CO and NOx emissions. While our models provided strong results in both regression and classification, the ELM better captures the complex patterns in the data, which further emphasizes the need for more advanced nonlinear models in emission prediction. Thus, for tasks invovling complex patterns in the data, ELM can offer significant advantages in capturing these intricate relationships.

# 5 Exam Problems

## 5.1 Question 3

There is one parameter for each connection within the network. As there are 7 input nodes connected to 10 nodes in the hidden layer, and 10 biases, one for each hidden unit, the total number of parameters between the input and hidden layer $7 * 10 + 10 = 80$. Between the hidden and output layer we have 4 classes connected to 10 hidden units giving $10 * 4 = 40$ with one bias pr output $= 4$ biases. In total, this gives

$$80 + 40 + 4 = 124$$

therefore, answer **A)** is correct.

## 5.2 Question 4

By assigning the congestion levels from the decision tree to the classification boundary, it can be deducted that **D**) is correct, as the nodes are correctly assigned when **A**) $b_1 \geq -0.76$, **B**) $b_2 \geq 0.03$, **C**) $b_1 \geq -0.16$, **D**) $b_2 \geq 0.01$.

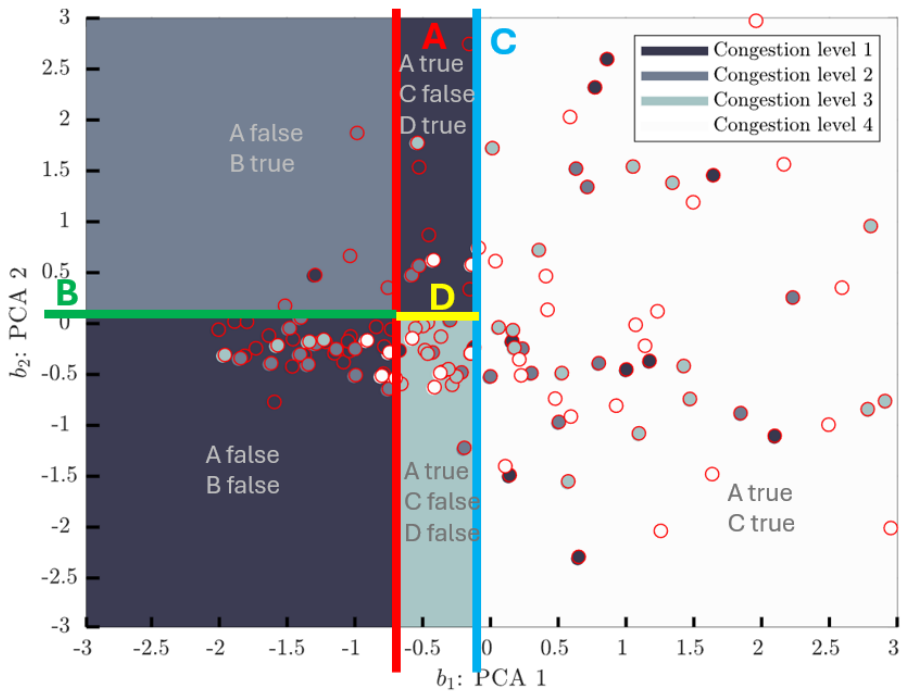


Figure 4: Classification boundary.

## 5.3 Question 5

There are

$$K_1(K_2 * n + 1) = 5 * (4 * 5 + 1) = 105$$

models to be trained in total, as there are 5 different values to be tested in each inner loop and the best model parameters are retrained on the whole inner loop data. By summing the time values for training and testing we can calculate the total time

$$105 * (20 + 5 + 8 + 1)ms = 3570ms$$

therefore, answer **C)** is correct.

## 5.4 Question 6

Using Python the probabilities for each observation has been calculated

```
import numpy as np

# Weights
w1 = np.array([1.2, -2.1, 3.2])
w2 = np.array([1.2, -1.7, 2.9])
w3 = np.array([1.3, -1.1, 2.2])

# Observations
a = np.array([1, -1.4, 2.6])
b = np.array([1, -0.6, -1.6])
c = np.array([1, 2.1, 5.0])
d = np.array([1, 0.7, 3.8])

# Probabilities
PA = 1 / (1 + np.exp(np.dot(a, w1)) + np.exp(np.dot(a, w2)) +
    np.exp(np.dot(a, w3)))
PB = 1 / (1 + np.exp(np.dot(b, w1)) + np.exp(np.dot(b, w2)) +
    np.exp(np.dot(b, w3)))
PC = 1 / (1 + np.exp(np.dot(c, w1)) + np.exp(np.dot(c, w2)) +
    np.exp(np.dot(c, w3)))
PD = 1 / (1 + np.exp(np.dot(d, w1)) + np.exp(np.dot(d, w2)) +
    np.exp(np.dot(d, w3)))

print("A =", PA, "B =", PB, "C =", PC, "D =", PD)

# Printed outcome
A = 3.0253229959961147e-06 B = 0.7304570363062249 C = 1.7674983200204553e-06
    D = 4.656384485887461e-06
```

Thus option **B**) show the highest probability of y = 4, and is the correct answer.

# 6 References

[1] Kaya, H., Tüfekci, P., & Uzun, E. (2019). Predicting CO and NOx emissions from gas turbines: Novel data and a benchmark PEMS. *Turkish Journal of Electrical Engineering and Computer Sciences*, *27*(6), 4783–4796. https://doi.org/10.3906/ELK-1807-87

# Appendix

## Appendix A: Impact of Number of Hidden Layers on Training and Test MSE

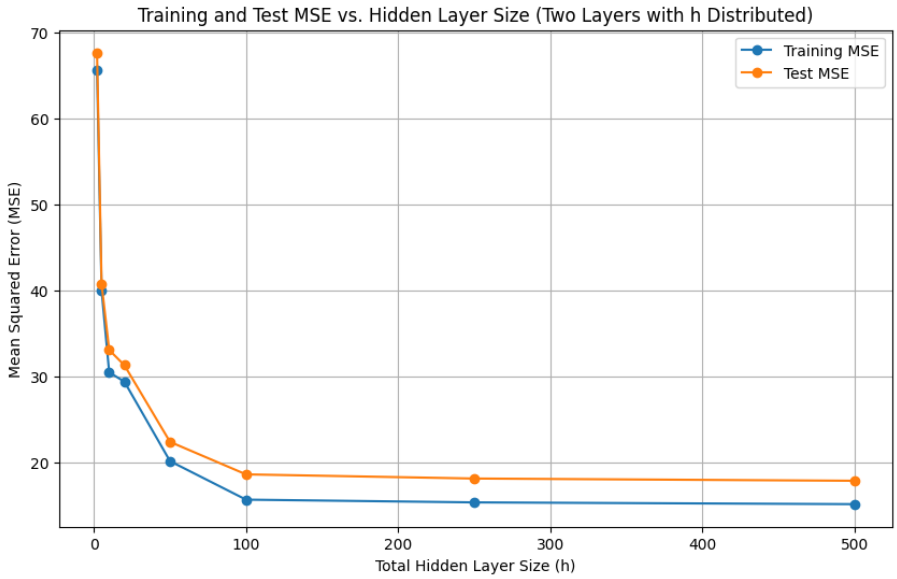


*Figure 4: Training and Test MSE for different h in two layer ANN*

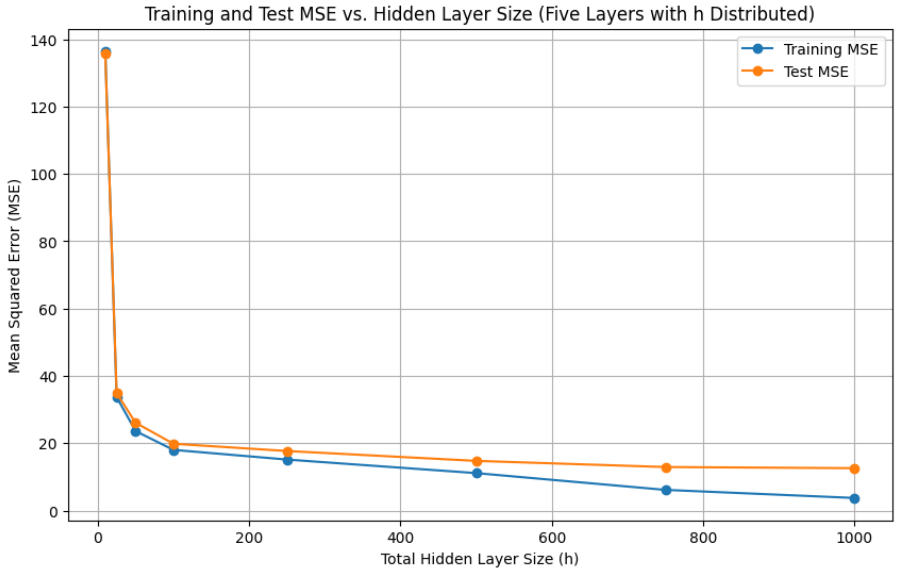


*Figure 5: Training and Test MSE for different h in five layer ANN*