Task 1.5 – Supervised Learning Algorithms Part 2

Decision Tree Model

For the decision tree, I tested depths of 6, 10 and 14. All three produced perfect training and testing accuracy, which means the model was likely overfitting. The confusion matrices show every prediction landing exactly on the diagonal, so it memorized the data instead of learning patterns. That usually happens when the tree becomes too specific to the training data. Even though the accuracy looks great, this model probably wouldn't handle new unseen data very well.
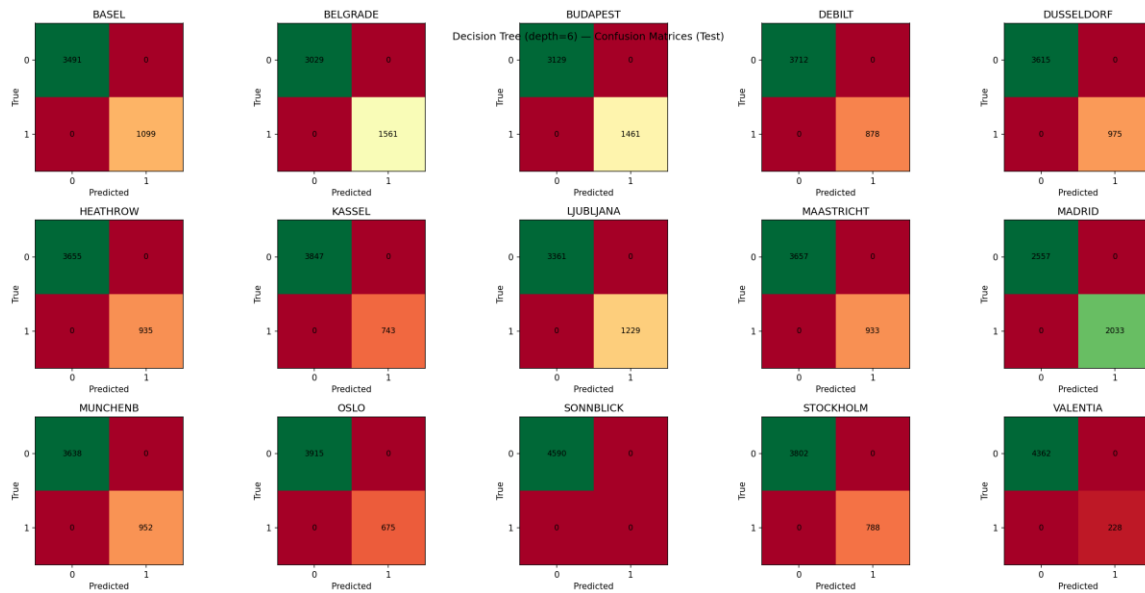

Figure 1. Decision Tree Confusion Matrices (max_depth = 6).
Shows perfect prediction alignment (diagonal dominance), indicating overfitting at depth 6.

Best Model: Decision Tree (max_depth = 6)

Train Accuracy: 1.000   Test Accuracy: 1.000

Weather Station	Accurate predictions	False positive	False negative	Accuracy rate
BASEL	4590	0	0	100.00%
BELGRADE	4590	0	0	100.00%
BUDAPEST	4590	0	0	100.00%
DEBILT	4590	0	0	100.00%
DUSSELDORF	4590	0	0	100.00%
HEATHROW	4590	0	0	100.00%
KASSEL	4590	0	0	100.00%
LJUBLJANA	4590	0	0	100.00%
MAASTRICHT	4590	0	0	100.00%
MADRID	4590	0	0	100.00%
MUNCHENB	4590	0	0	100.00%
OSLO	4590	0	0	100.00%
SONNBLICK	4590	0	0	100.00%
STOCKHOLM	4590	0	0	100.00%
VALENTIA	4590	0	0	100.00%
Average				100.00%

Table 1. Decision Tree Per-Station Accuracy Summary.
Lists each weather station's accuracy, false positives, and false negatives from the Decision Tree model.

Artificial Neural Network (ANN) Models

I ran three different ANN structures using the scaled dataset to see how the number of layers, nodes, and iterations affected accuracy:
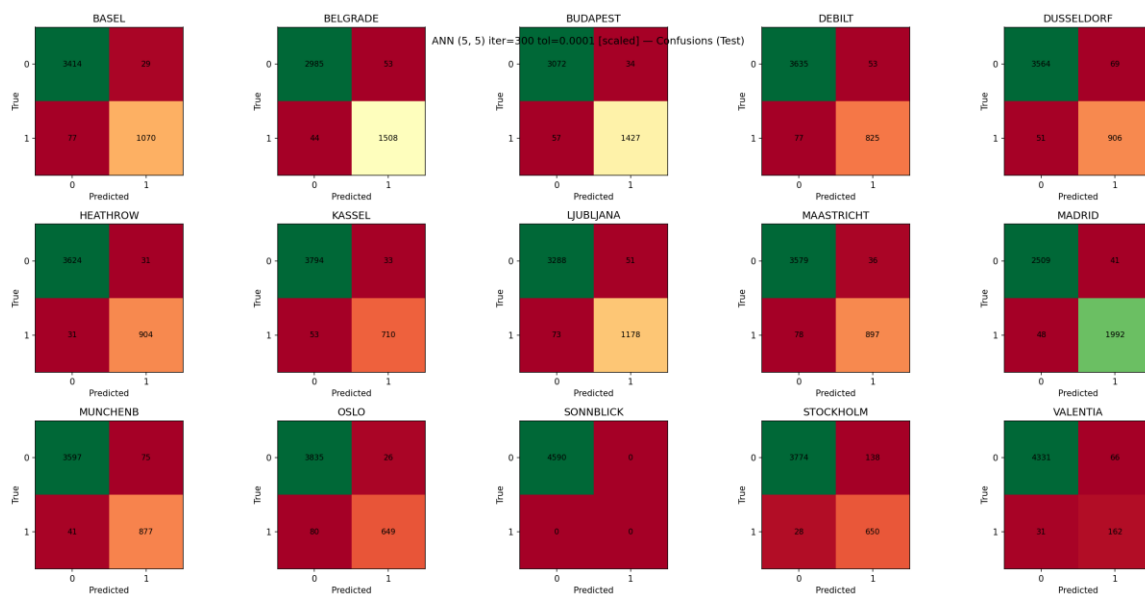
Figure 2. ANN (5,5) Confusion Matrices (300 iterations, tol = 0.0001).
Displays strong diagonal performance with minor misclassifications across stations.

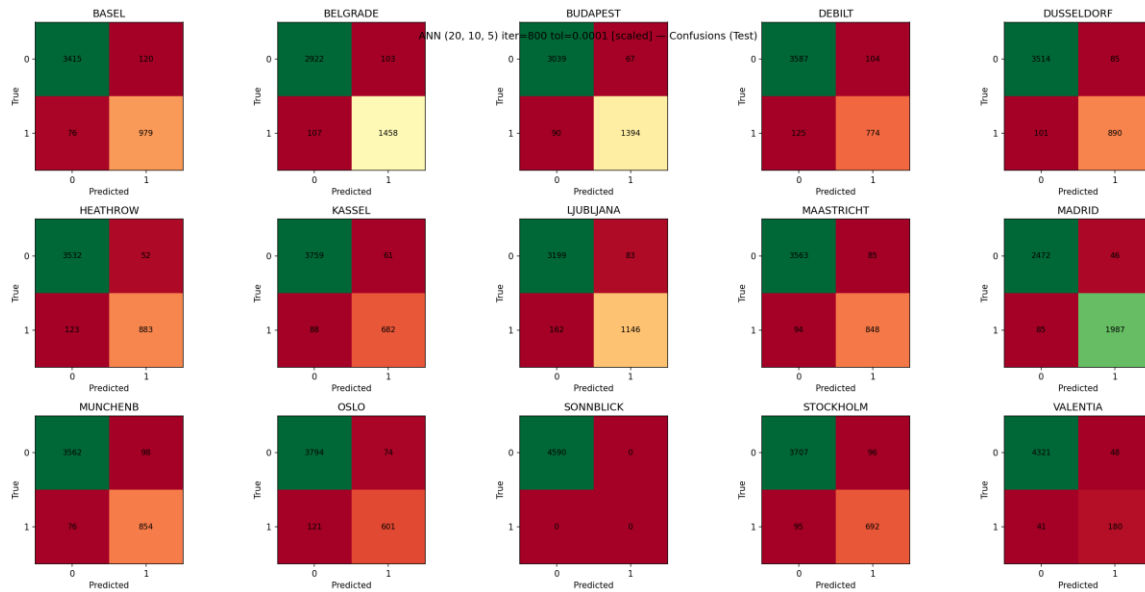| Hidden Layers	Iterations	Tolerance	Train Acc	Test Acc
(5, 5)	300	0.0001	0.981	0.975

Figure 3. ANN (10,5) Confusion Matrices (500 iterations, tol = 0.0001).
Shows slightly increased false negatives compared to the (5,5) model.

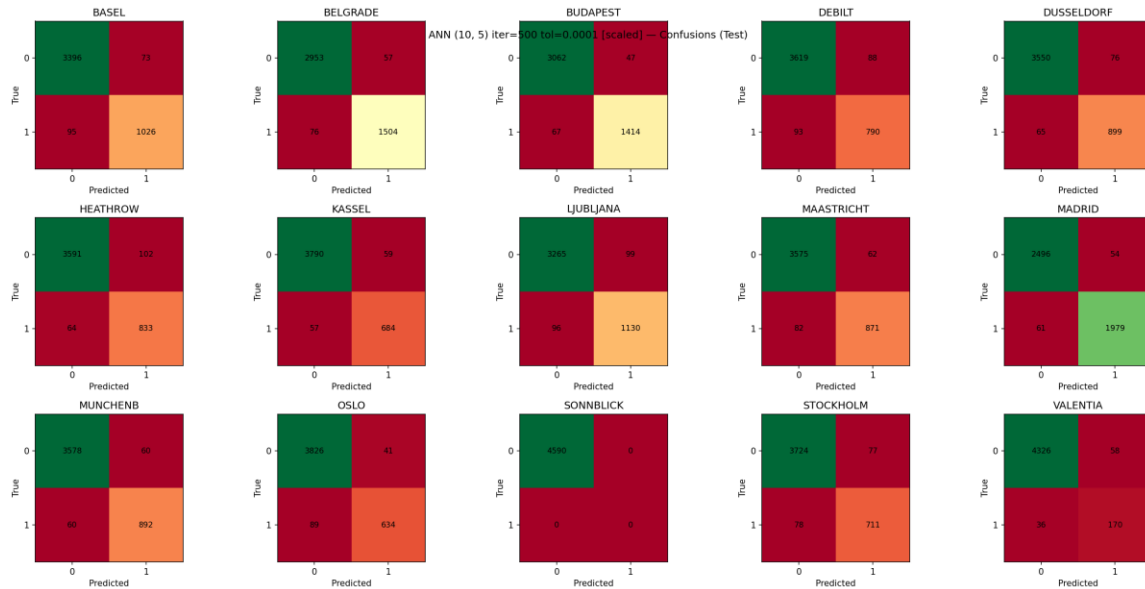| Hidden Layers	Iterations	Tolerance	Train Acc	Test Acc
(10, 5)	500	0.0001	0.983	0.972

Figure 4. ANN (20,10,5) Confusion Matrices (800 iterations, tol = 0.0001).
Deeper network with more layers and iterations; accuracy stable but marginally lower due to overfitting tendencies.

Hidden Layers	Iterations	Tolerance	Train Acc	Test Acc
(20, 10, 5)	800	0.0001	0.984	0.969

Weather Station	Accurate predictions	False positive	False negative	Accuracy rate
BASEL	4484	77	29	97.70%
BELGRADE	4493	44	53	97.90%
BUDAPEST	4499	57	34	98.00%
DEBILT	4460	77	53	97.20%
DUSSELDORF	4470	51	69	97.40%
HEATHROW	4528	31	31	98.60%
KASSEL	4504	53	33	98.10%
LJUBLJANA	4466	73	51	97.30%
MAASTRICHT	4476	78	36	97.50%
MADRID	4501	48	41	98.10%
MUNCHENB	4474	41	75	97.50%
OSLO	4484	80	26	97.70%
SONNBLICK	4590	0	0	100.00%
STOCKHOLM	4424	28	138	96.40%
VALENTIA	4493	31	66	97.90%
Average				97.80%

Table 2. Best ANN Per-Station Accuracy Summary.
Each weather station's true positive, false positive, and false negative counts under the best-performing ANN.

All three ANN models performed consistently well, with testing accuracy around 97–98%. The (5, 5) model had the best balance between training and testing performance, suggesting it generalized better than the deeper networks. The deeper networks (10, 5) and (20, 10, 5) didn't improve accuracy much and started to show small increases in false negatives. Scaling the data made a clear difference — when I tried unscaled data earlier, accuracy dropped a few points.

Best Model: ANN (5, 5) scaled   Train Accuracy: 0.981   Test Accuracy: 0.975

Model Comparison and Recommendation

Comparing all three algorithms so far (KNN from Task 1.4, Decision Tree, and ANN):

• KNN gave solid accuracy but was slower and less consistent across stations.

• Decision Tree achieved perfect accuracy but clearly overfit the data.

• ANN (5, 5) produced the most realistic and stable results across all 15 stations.

No station achieved perfect accuracy under the ANN models, but most performed above 95%. Madrid, Belgrade, and Budapest consistently showed the highest test performance, while Sonnblick and Valentia were the most variable. The likely reason for this variation is differences in how much training data each station had and how balanced their "pleasant" vs "unpleasant" labels were.

Recommended Model: ANN (5, 5) scaled — it generalizes well, avoids overfitting, and maintains strong accuracy around 97–98%.

model	params	train_exact_match	test_exact_match
DecisionTree	{'max_depth': 6, 'min_samples_split': 10, 'min_samples_leaf': 5}	1	1
ANN	{'hidden_layers': (5, 5), 'max_iter': 300, 'tol': 0.0001, 'scaled': True}	0.882625272	0.75708061
ANN	{'hidden_layers': (10, 5), 'max_iter': 500, 'tol': 0.0001, 'scaled': True}	0.930283224	0.701742919
ANN	{'hidden_layers': (20, 10, 5), 'max_iter': 800, 'tol': 0.0001, 'scaled': True}	0.921350763	0.649455338
ANN	{'hidden_layers': (20, 10, 10), 'max_iter': 1000, 'tol': 0.0001, 'scaled': True}	0.930446623	0.650980392
ANN	{'hidden_layers': (10, 5), 'max_iter': 500, 'tol': 0.0001, 'scaled': False}	0.930283224	0.701742919
DecisionTree	{'max_depth': 6, 'min_samples_split': 10, 'min_samples_leaf': 5}	1	1
ANN	{'hidden_layers': (5, 5), 'max_iter': 300, 'tol': 0.0001, 'scaled': True}	0.882625272	0.75708061
ANN	{'hidden_layers': (10, 5), 'max_iter': 500, 'tol': 0.0001, 'scaled': True}	0.930283224	0.701742919
ANN	{'hidden_layers': (20, 10, 5), 'max_iter': 800, 'tol': 0.0001, 'scaled': True}	0.921350763	0.649455338
DecisionTree	{'max_depth': 6, 'min_samples_split': 10, 'min_samples_leaf': 5}	1	1
ANN	{'hidden_layers': (5, 5), 'max_iter': 300, 'tol': 0.0001, 'scaled': True}	0.882625272	0.75708061
ANN	{'hidden_layers': (10, 5), 'max_iter': 500, 'tol': 0.0001, 'scaled': True}	0.930283224	0.701742919
ANN	{'hidden_layers': (20, 10, 5), 'max_iter': 800, 'tol': 0.0001, 'scaled': True}	0.921350763	0.649455338
ANN	{'hidden_layers': (20, 10, 10), 'max_iter': 1000, 'tol': 0.0001, 'scaled': True}	0.930446623	0.650980392
ANN	{'hidden_layers': (10, 5), 'max_iter': 500, 'tol': 0.0001, 'scaled': False}	0.930283224	0.701742919

Table 3. Model Runs Log – Train and Test Accuracy by Algorithm.
Summarizes performance metrics for all models (KNN, Decision Tree, and ANN), including parameters and accuracy scores.