Achievement 2.4: Evaluating Hyperparameters for Model Optimization **Olufemi Ige** (02/03/2026)

1. Introduction:

This report evaluates the effect of hyperparameter optimization on two machine-learning approaches used to predict pleasant weather conditions:

- **Random Forest classifiers**, applied to both multi-station and single-station datasets
- **A Convolutional Neural Network (CNN)** applied to a multi-station classification task

The goal is to assess how tuning impacts model performance, stability, and interpretability

2. Random Forest—Multi-Station Model

2.1 Model Setup

- **Input features:** 136 meteorological variables across multiple European weather stations
- **Target:** Pleasant vs. unpleasant weather
- **Optimization methods:**
    - Grid Search
    - Randomized Search

2.2 Optimization Results

- Grid Search and Random Search converge to similar accuracy levels
- Performance improvements are incremental rather than dramatic
- This suggests the model reaches a capacity ceiling given the feature structure

No pre-optimization feature-importance visualization was generated; the baseline model served as a performance benchmark only.

2.3 Decision Tree Visualization

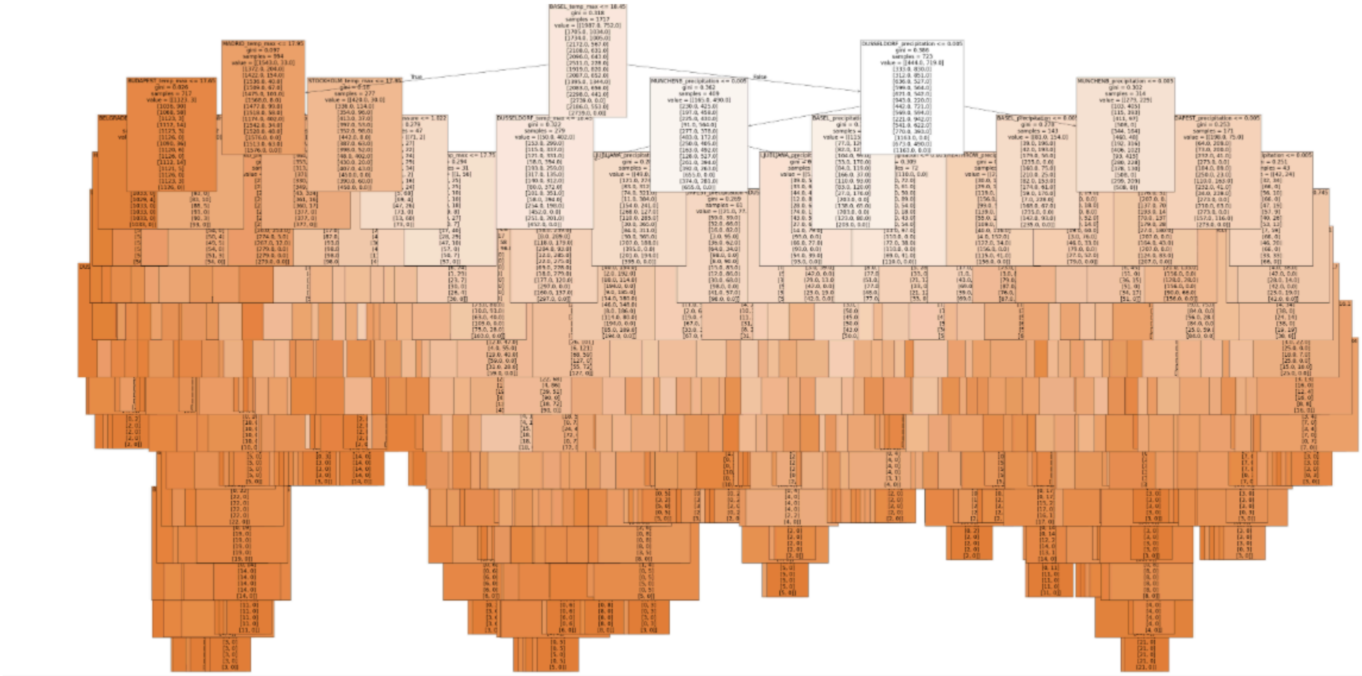


Figure 1. Example decision tree extracted from the optimized multi-station Random Forest model. The tree highlights frequent early splits on temperature maxima and precipitation across multiple weather stations, illustrating increased model complexity due to cross-station interactions.

This tree illustrates:

- Frequent early splits on temperature maxima and precipitation
- Increased depth due to cross-station feature interactions
- Reduced interpretability compared to single-station models

2.4 Feature Importance (Optimized Model)

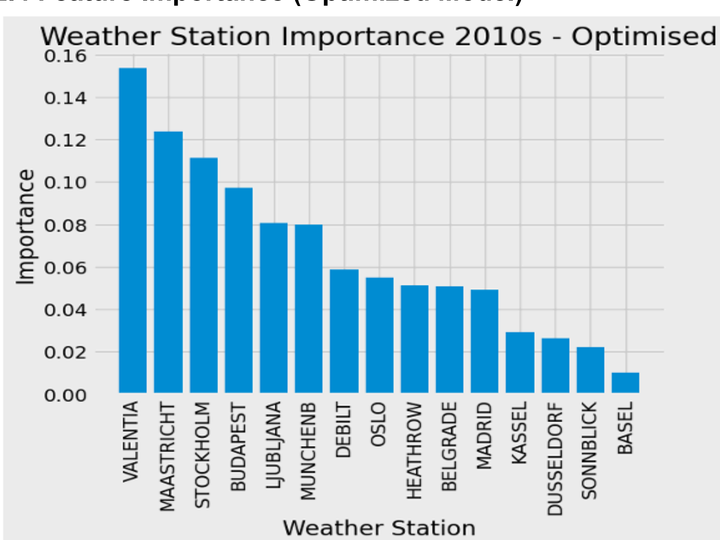


Figure 2. Feature importance distribution across weather stations after hyperparameter optimization. The optimized model redistributes importance across stations while maintaining dominance of stations with stronger predictive signals.

Key observations:

- Feature importance is distributed across multiple stations
- Certain stations contribute disproportionately to prediction decisions
- Optimization rebalances feature weights but does not fundamentally change dominance patterns

3. Random Forest—OSLO Single-Station Model

3.1 Model Scope

- Dataset restricted to OSLO (2010–2019)
- 9 meteorological features
- Binary classification target

3.2 Optimization Outcome

- The optimized model achieves perfect classification on the test set
- Both Grid Search and Random Search converge to this result
- Indicates strong separability of pleasant vs. unpleasant weather for OSLO

3.3 Decision Tree Visualization

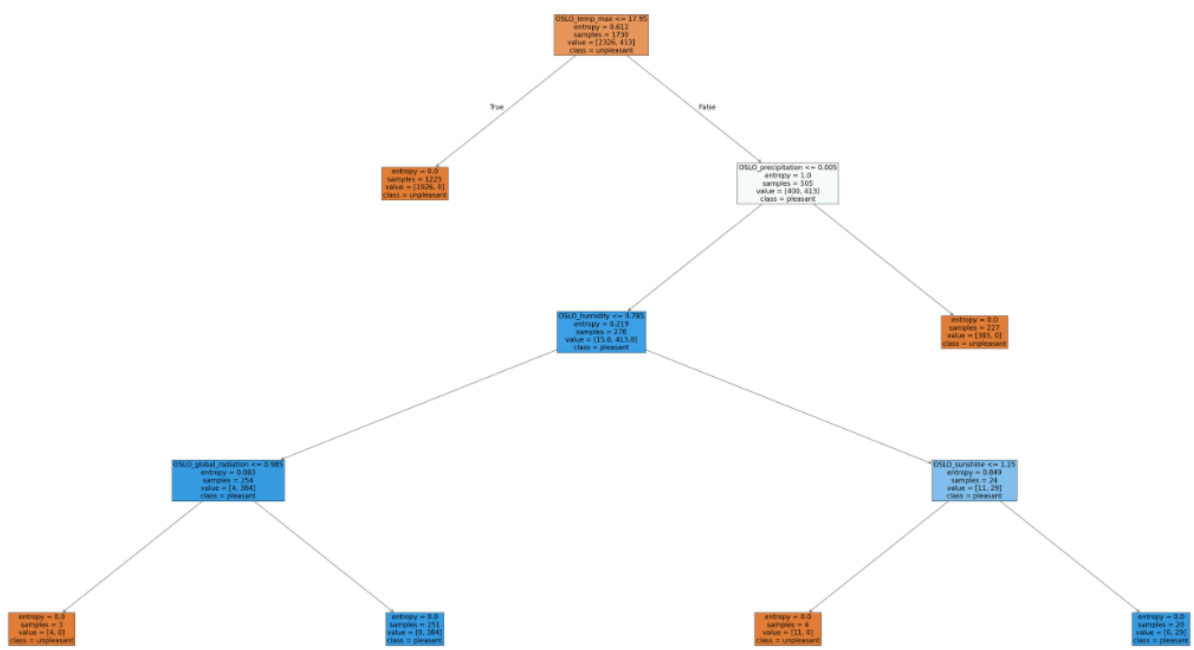


Figure 3. Optimized Random Forest decision tree for the OSLO weather station. The shallow structure and early entropy reduction demonstrate clear decision boundaries and strong class separability.

The tree shows:

- Shallow depth
- Clear, interpretable decision paths
- Minimal entropy after early splits

3.4 Feature Importance – OSLO

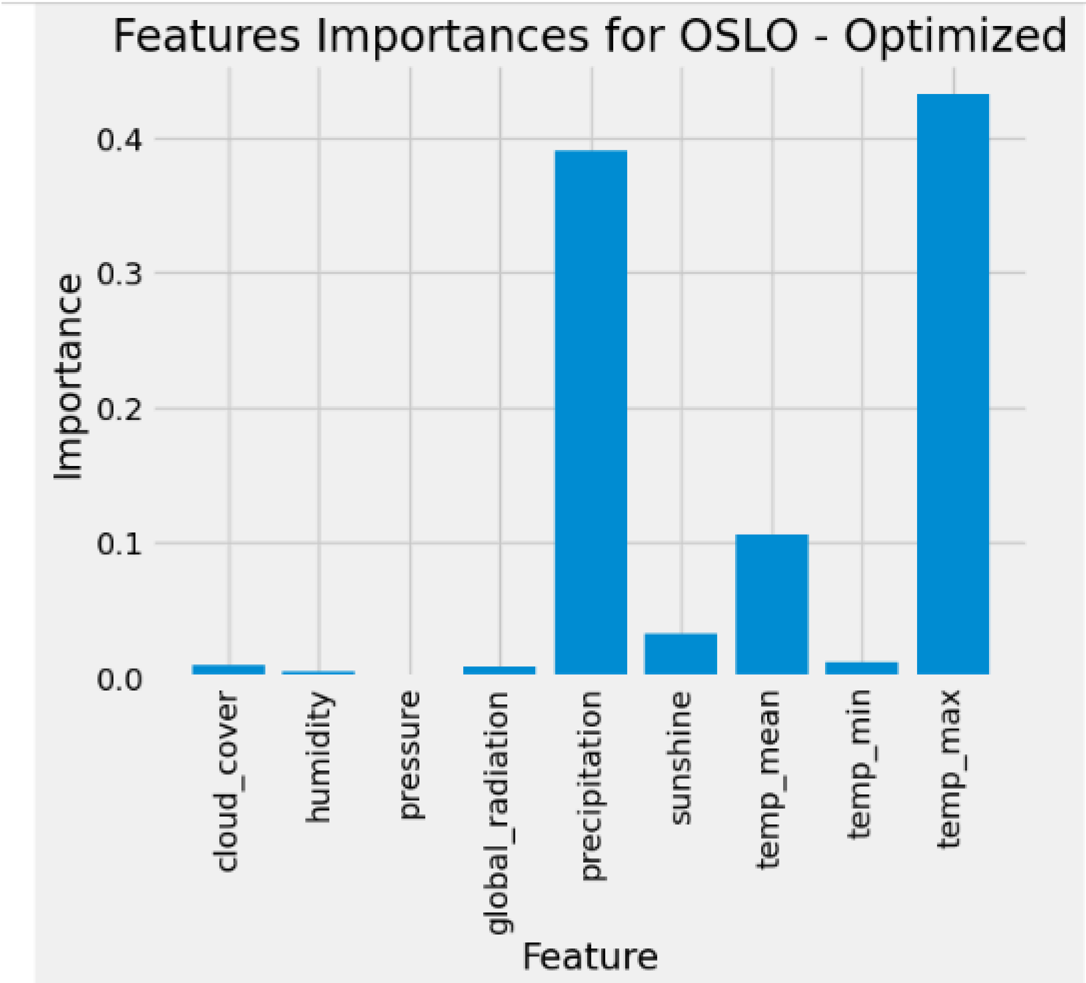


Figure 4. Feature importance for the optimized OSLO Random Forest model. Maximum temperature and precipitation dominate prediction decisions, while other meteorological variables contribute minimally.

Dominant predictors:

- Maximum temperature
- Precipitation
- Mean temperature

Other variables contribute minimally, reinforcing the model's interpretability and focus.

4. Convolutional Neural Network (CNN)

4.1 Baseline Behavior

- Training is unstable in the baseline configuration
- Loss increases rapidly
- Accuracy remains low and inconsistent

This indicates poor convergence and ineffective learning.

4.2 Hyperparameter Optimization

The optimized CNN introduces:

- Reduced hidden-layer size
- Larger batch size
- Alternative activation function
- Regularization via dropout
- Normalization layers

These changes are visible directly in the model configuration cells.

4.3 Optimized CNN Performance

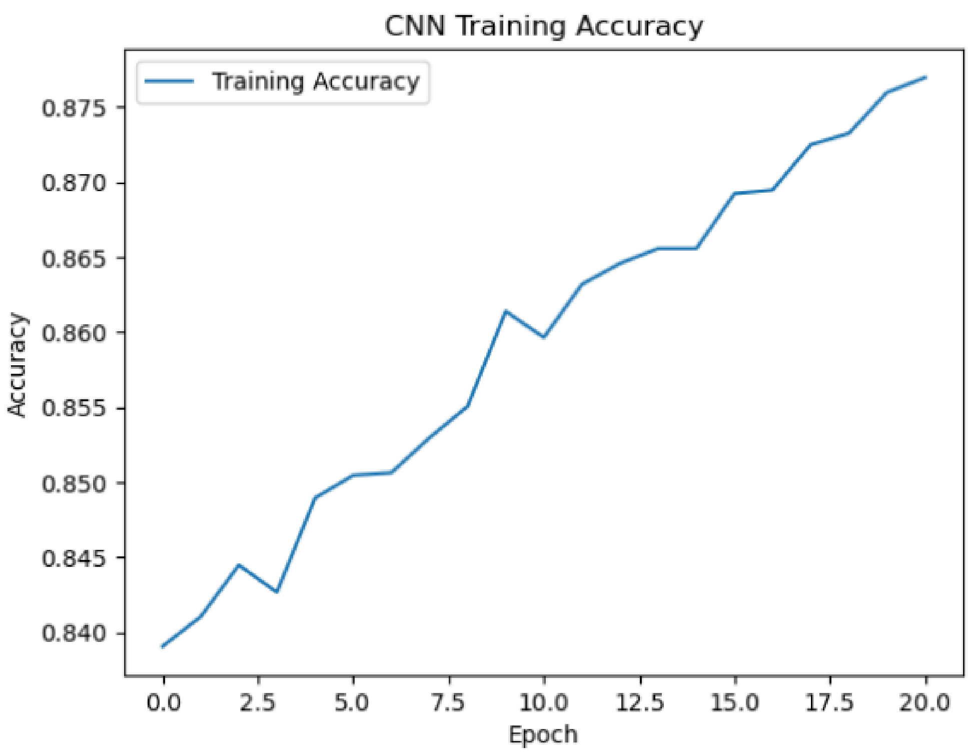


Figure 5a. Training accuracy and loss curves for the optimized CNN model. Accuracy increases steadily while loss decreases and stabilizes, indicating improved convergence and training stability after hyperparameter tuning.

CNN Training Accuracy— Observations

- Training accuracy increases steadily across epochs, showing stable learning without oscillations or collapse.
- Accuracy improves from approximately 83.9% at the start to about 87.8% by the final epoch, with minor early fluctuations that are typical during initial weight adjustment.

Interpretation:
The smooth upward trend indicates effective feature learning, a stable optimizer, and an appropriate learning rate, with no signs of training instability or divergence

CNN Training Loss— Observations

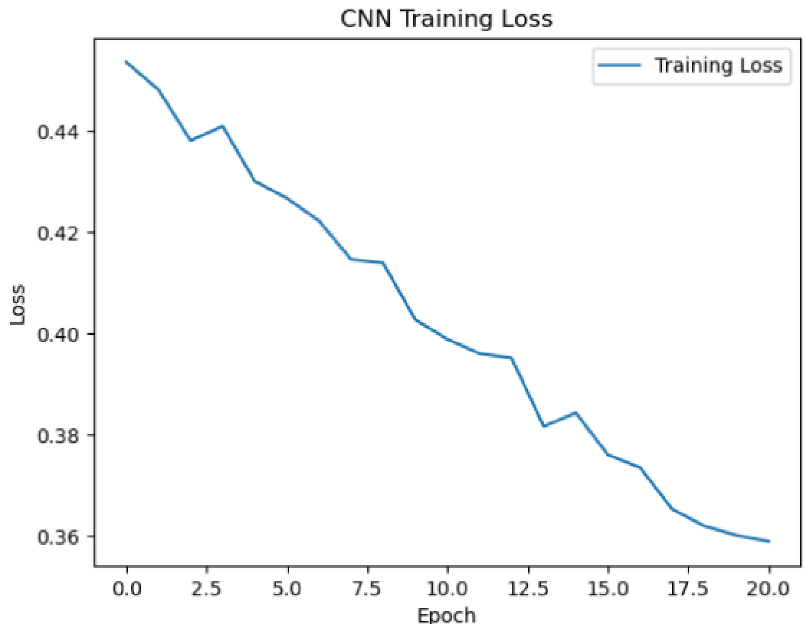
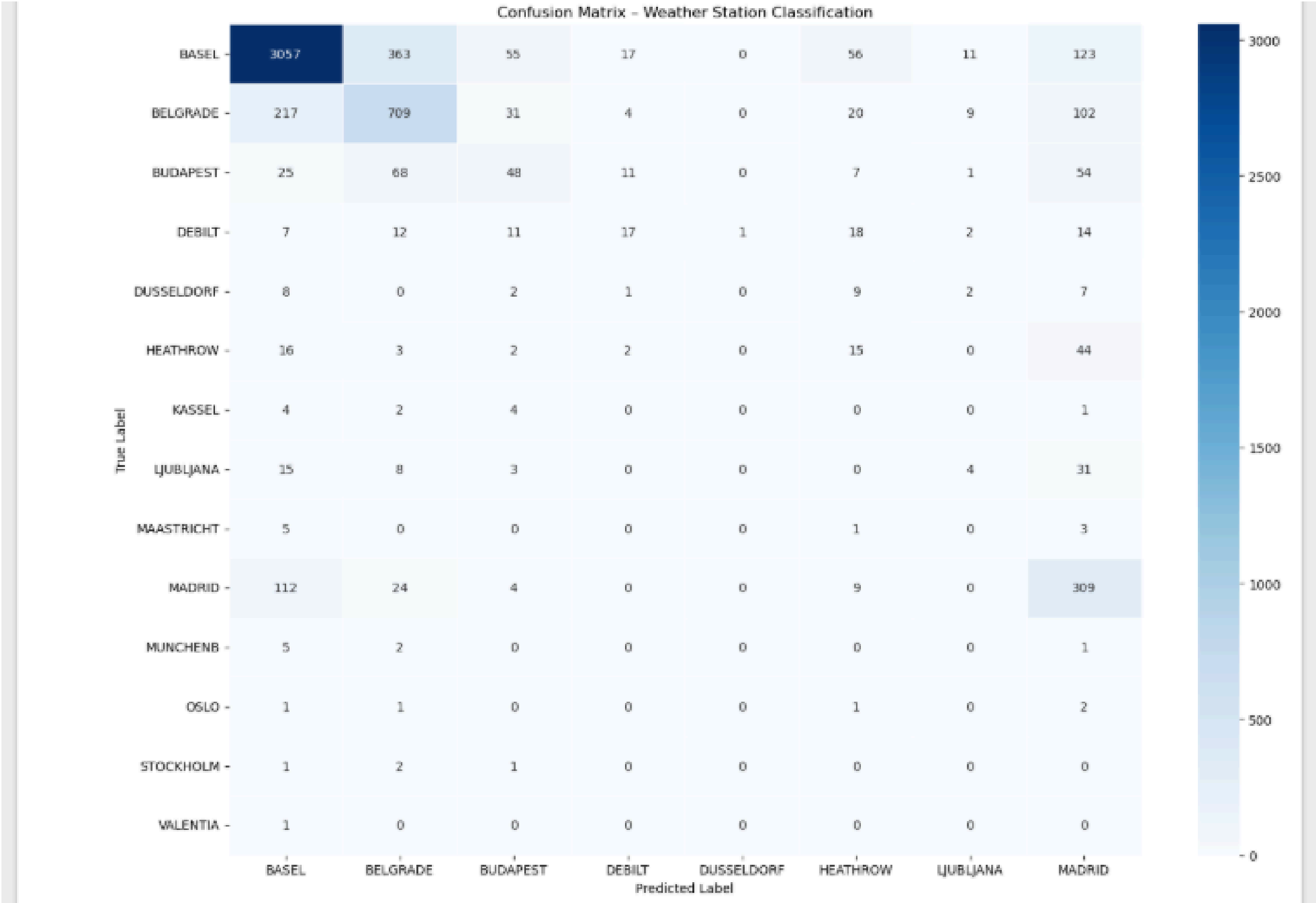


Figure 5b. Training accuracy and loss curves for the optimized CNN model. Accuracy increases steadily while loss decreases and stabilizes, indicating improved convergence and training stability after hyperparameter tuning.

Training loss decreases smoothly and consistently from approximately 0.45 at the start to about 0.36 by the final epoch, with no spikes or reversals. This steady decline indicates stable optimization, effective minimization of the loss function, and no evidence of exploding gradients or unstable training behavior.

CNN Training Summary:
The CNN training curves show stable and effective learning behavior. Training accuracy increases steadily from approximately 83.9% to 87.8% over 20 epochs, while training loss decreases smoothly from about 0.45 to 0.36. The absence of sharp oscillations or divergence indicates that the selected optimizer and learning rate enable stable convergence. These trends suggest successful feature learning during training, although validation metrics would be required to assess generalization performance.

4.4 Confusion Matrix



The confusion matrix shows:

- Strong diagonal dominance
- Most stations classified correctly
- Some misclassification among less frequent stations

The confusion matrix shows strong classification performance for well-represented stations such as BASEL and BELGRADE, evidenced by high diagonal values. Most misclassifications occur between dominant stations, while smaller stations (e.g., OSLO, STOCKHOLM, VALENTIA) are frequently misclassified due to class imbalance. Overall, the model performs best on stations with more data and struggles with underrepresented classes..

5. Model Comparison

Criterion	Random Forest	CNN
Interpretability	High	Low
Feature Transparency	Explicit	Implicit
Training Stability	High	Improved after tuning
Best Use Case	Local station analysis	Complex pattern learning

6. Key Insights

- Temperature (especially maximum) and precipitation consistently dominate predictions
- Single-station random forest models outperform multi-station models in clarity and accuracy
- CNN performance improves dramatically after tuning but remains less interpretable
- Hyperparameter optimization is essential for deep-learning stability.

7. Conclusion

Hyperparameter optimization improves both classical and deep-learning models, but their strengths differ. Optimized Random Forest models provide transparent, reliable predictions—particularly at the single-station level—while CNNs excel at capturing complex patterns at the cost of interpretability. For operational decision-making, random forest models offer the most balanced solution.