Computational Quantum Physics & Applications: Classification for Higgs Signal vs. Background

ELEFTHERIOS MARIOS ZOGRAFOS
AEM: 4428

1 Data Preprocessing

- **Load dataset**: `HIGGS_8K.csv` with 8 000 samples.
- **Feature subsets**:
  - *Complete*: all 28 input variables.
  - *Low-level*: first 21 raw features.
  - *High-level*: last 7 derived variables.
- **Train/test split**: stratified 75%/25% split to preserve signal/background ratio.

2 Classification Methods

Applied to each feature set:

1. **k-Nearest Neighbors (kNN)**: grid search over $k = 1 \dots 249$ (default `max_neighbors=250`), 5-fold CV, scoring=`roc_auc`.
2. **Decision Tree**: grid search over
   - `criterion` = {`gini, entropy, log_loss`},
   - `max_depth` = {None, 10, 20, 30},
   - `min_samples_split` = {2, 5, 10},
   - `min_samples_leaf` = {1, 2, 4},

   with 5-fold CV optimizing ROC AUC (`scoring='roc_auc'`).
3. **Random Forest**: grid search over
   - `n_estimators` = {50, 100, 200},
   - `criterion` = {`gini, entropy`},
   - `max_depth` = {None, 10, 20, 30},
   - `min_samples_split` = {2, 5, 10},
   - `min_samples_leaf` = {1, 2, 4},

   using 5-fold CV on ROC AUC.

4. **Artificial Neural Network (ANN)**:
   - Architecture: Dense layers with [256, 128, 64, 32, 16] units
   - Activations: ELU + BatchNormalization + Dropout rates [0.3, 0.3, 0.3, 0.2, 0.2]
   - Output: Dense(1, sigmoid)
   - Loss: binary cross-entropy; Optimizer: Adam (lr=1e-3)
   - Callbacks: EarlyStopping(`monitor='val_auc'`, patience=10, restore_best_weights=True), ReduceLROnPlateau(`monitor='val_auc'`, factor=0.5, patience=5, min_lr=1e-6)
   - Training: up to 500 epochs, batch_size=256, `verbose=0`

3 Performance Summary

Accuracy and AUC for each method & feature set:

Method	**Complete (Acc/AUC)**	**Low-level (Acc/AUC)**	**High-level (Acc/AUC)**
kNN	0.5957 / 0.6645	0.5712 / 0.6113	0.6767 / 0.7497
Decision Tree	0.6522 / 0.6852	0.5522 / 0.5686	0.6547 / 0.6973
Random Forest	0.7031 / 0.7833	0.5947 / 0.6374	0.6867 / 0.7651
ANN	0.7026 / 0.7691	0.5922 / 0.6334	0.6872 / 0.7630

Table 1: Accuracy and AUC for each classification method across feature sets.

4 Conclusions

Feature-set-based Findings

- **Low-level quantities (21 features)**: AUC in [0.57–0.64], Accuracy = 0.55-0.60. *Poor discrimination on low-level inputs.*
- **High-level quantities (7 features)**: AUC in [0.75–0.81], Accuracy =0.68. *Provide strong signal/background separation.*
- **Complete feature set (28 features)**: AUC in [0.78–0.80], Accuracy =0.70–0.71. *Combining low- and high-level features yields the best overall performance.*

Model-level Trade-offs

Model	AUC	Acc.	Runtime	Interpretability
kNN	Moderate	Moderate	Very fast ($\leq$ 2.5 min)	Low (lazy, nonparametric)
Decision Tree	Moderate–Good	Moderate	Very fast ($\leq$ 1.5 min)	High (simple rules)
Random Forest	Best (0.78)	Best (0.71)	Slow ($\approx$ 32 min)	Medium (ensemble)
ANN	Near-RF (0.77)	Near-RF (0.70)	Very Fast ($\leq$ 1.5 min)	Low (black box)

In this report, we demonstrated that classical classifiers (kNN, Decision Tree, Random Forest) and a custom ANN can effectively distinguish Higgs signal from background, showing that high-level features drive the strongest separation and that the ANN offers the best balance of accuracy, AUC, and computational efficiency.

Computational Resources & Libraries

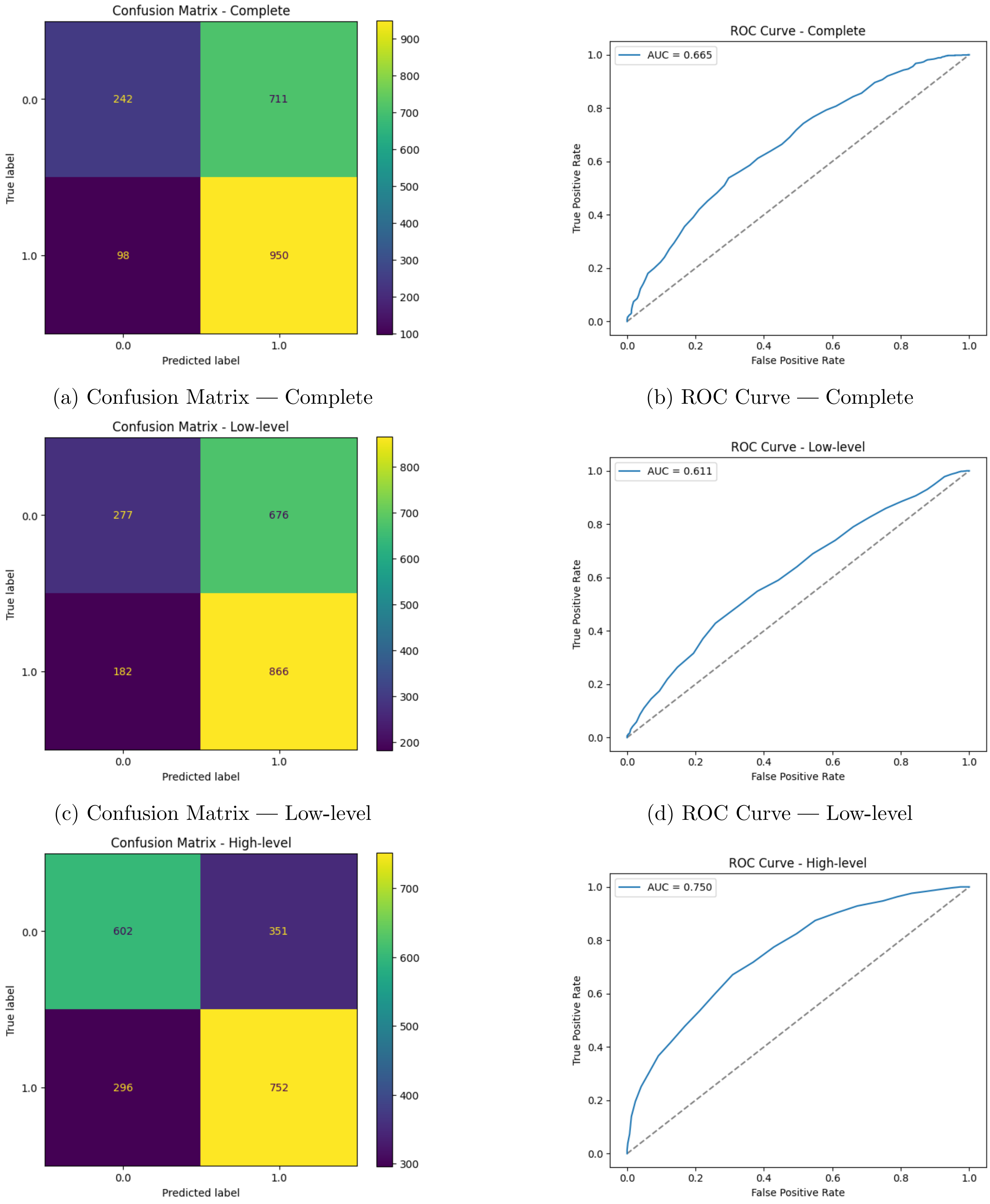
All code was written in Python using the following libraries:

- numpy
- pandas
- matplotlib
- joblib
- sklearn
- tensorflow
- scipy

The computations were performed on a system with an Intel® Core™ i7-8750H CPU @ 2.20 GHz and 16 GB of DDR4 RAM.

Appendix - Supplementary Figures

kNN: Confusion Matrices and ROC Curves



(a) Confusion Matrix — Complete

(b) ROC Curve — Complete

(c) Confusion Matrix — Low-level

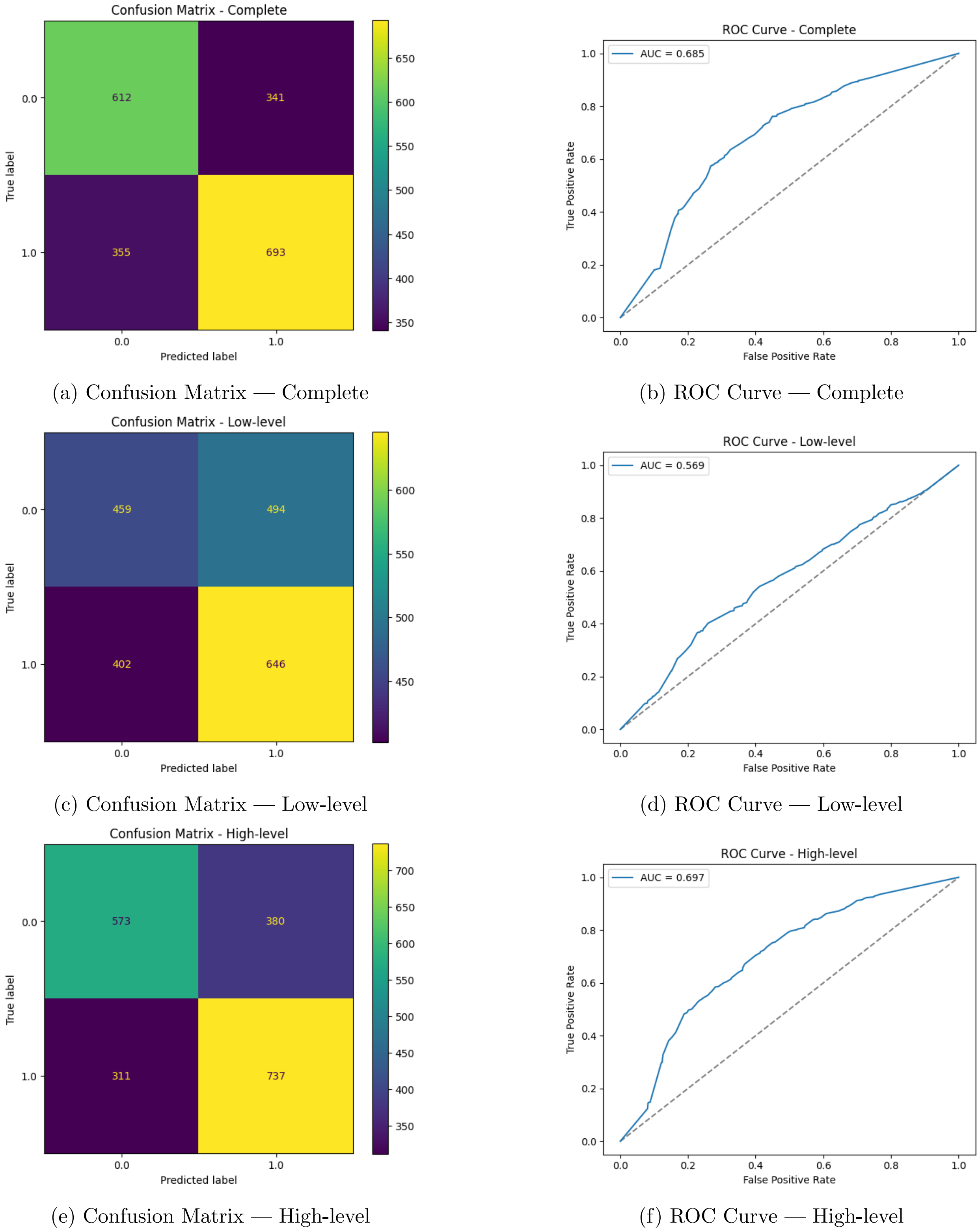
(d) ROC Curve — Low-level

(e) Confusion Matrix — High-level

(f) ROC Curve — High-level

Figure 1: Performance of kNN classifier on each feature set.

Decision Tree: Confusion Matrices and ROC Curves



(a) Confusion Matrix — Complete

(b) ROC Curve — Complete

(c) Confusion Matrix — Low-level

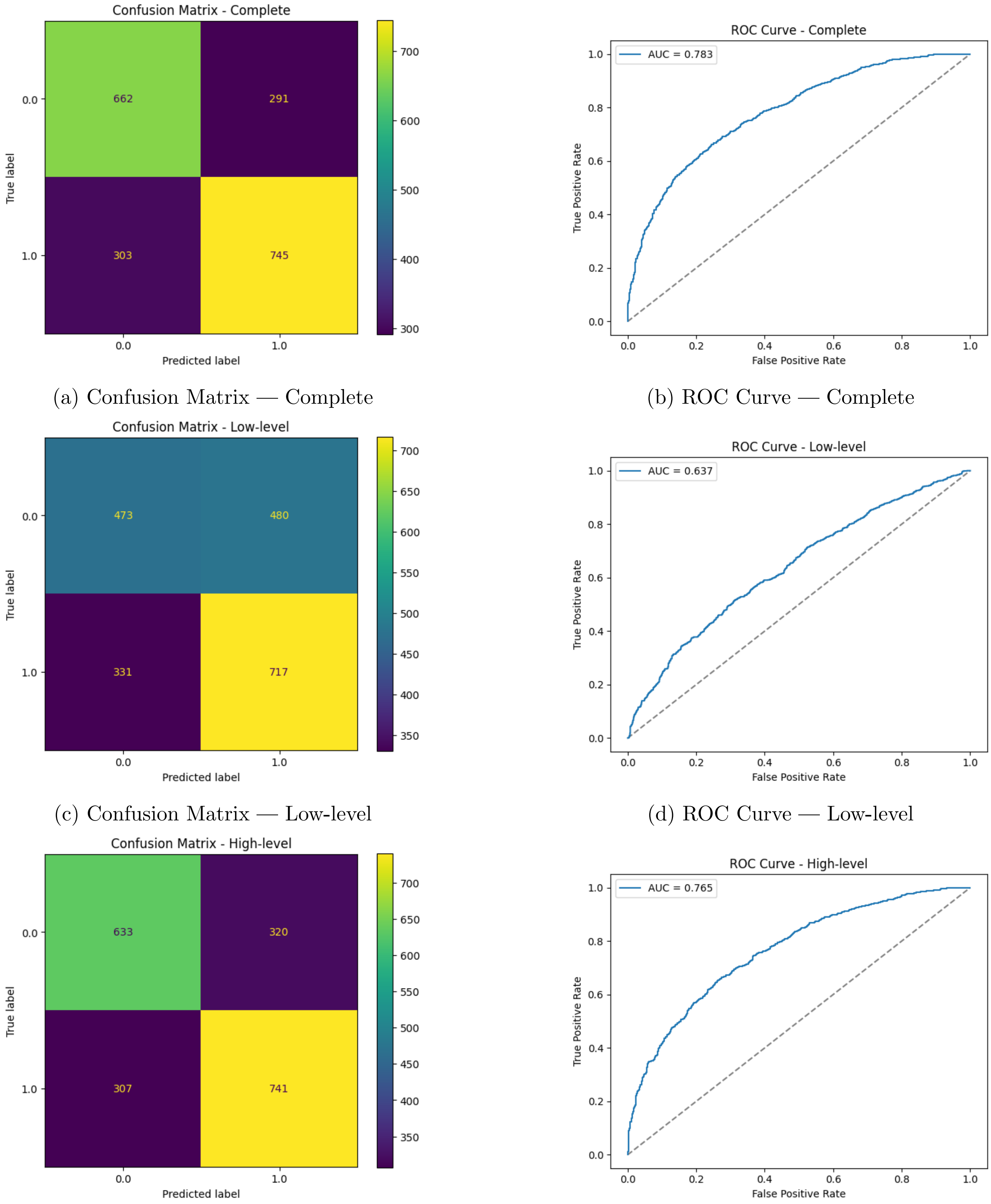
(d) ROC Curve — Low-level

(e) Confusion Matrix — High-level

(f) ROC Curve — High-level

Figure 2: Performance of Decision Tree classifier on each feature set.

Random Forest: Confusion Matrices and ROC Curves



(a) Confusion Matrix — Complete

(b) ROC Curve — Complete

(c) Confusion Matrix — Low-level

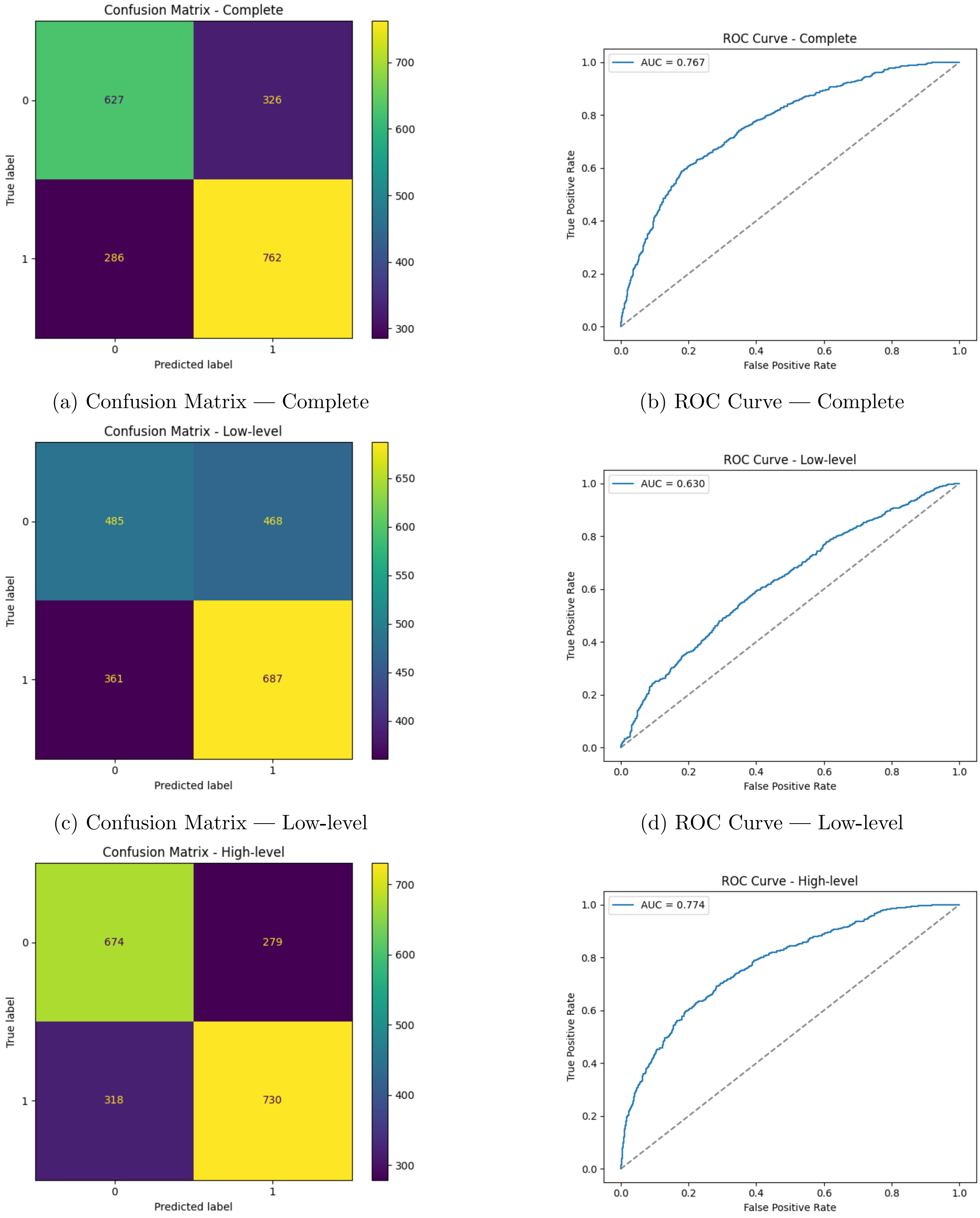
(d) ROC Curve — Low-level

(e) Confusion Matrix — High-level

(f) ROC Curve — High-level

Figure 3: Performance of Random Forest classifier on each feature set.

ANN: Confusion Matrices and ROC Curves



(a) Confusion Matrix — Complete

(b) ROC Curve — Complete

(c) Confusion Matrix — Low-level

(d) ROC Curve — Low-level

(e) Confusion Matrix — High-level

(f) ROC Curve — High-level

Figure 4: Performance of ANN on each feature set.