

## Model Optimization and Tuning Phase

Date	19 Feb 2026
Team ID	LTVIP2026TMIDS80731
Project Title	Online Payment Fraud Detection using ML
Maximum Marks	10 Marks

### Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

#### Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Random Forest	<pre>param_dist = {     'n_estimators': [50, 100],     'max_depth': [None, 10],     'class_weight': ['balanced'] }  rf_clf = RandomForestClassifier(random_state=42)</pre>	<pre>Best Parameters: {'n_estimators': 50, 'max_depth': None, 'class_weight': 'balanced'} Best Cross-Validation Score: 0.8434</pre>
Decision Tree	<pre>dt_param_dist = {     'max_depth': [None, 10, 15],     'min_samples_split': [2, 10],     'min_samples_leaf': [1, 5],     'criterion': ['gini', 'entropy'],     'class_weight': ['balanced', {0: 1, 1: 10}] }  dt_clf = DecisionTreeClassifier(random_state=42)</pre>	<pre>Best Parameters: {'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': None, 'criterion': 'gini', 'class_weight': {0: 1, 1: 10}} Best Cross-Validation Score: 0.8260</pre>

KNN	<pre> knn_param_dist = {     'n_neighbors': [3, 5, 7],     'weights': ['uniform', 'distance'],     'metric': ['euclidean', 'manhattan'] }  knn_clf = KNeighborsClassifier() </pre>	<pre> Best Parameters: {'weights': 'distance', 'n_neighbors': 5, 'metric': 'manhattan'} Best Cross-Validation Score: 0.7024 </pre>
Gradient Boosting	<pre> gb_param_dist = {     'n_estimators': [50, 100],     'learning_rate': [0.05, 0.1],     'max_depth': [3, 5],     'subsample': [0.8, 1.0] }  gb_clf = GradientBoostingClassifier(random_state=42) </pre>	<pre> Best Parameters: {'subsample': 1.0, 'n_estimators': 50, 'max_depth': 3, 'learning_rate': 0.05} Best Cross-Validation Score: 0.6604 </pre>

### Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric
Random Forest	<pre> Optimized Random Forest Evaluation: Classification Report:       precision    recall  f1-score   support           0       1.00     1.00     1.00   1270881           1       0.98     0.76     0.85     1643        accuracy                           0.9997       macro avg       0.99     0.88     0.93   1272524       weighted avg    1.00     1.00     1.00   1272524  Accuracy: 0.9997 Confusion Matrix: [[1270856      25]  [     400     1243]] </pre>

	<p><b>Optimized Decision Tree Evaluation:</b></p> <p><b>Classification Report:</b></p> <table> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>1.00</td><td>1.00</td><td>1.00</td><td>1270881</td></tr> <tr> <td>1</td><td>0.87</td><td>0.83</td><td>0.85</td><td>1643</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>1.00</td><td>1272524</td></tr> <tr> <td>macro avg</td><td>0.93</td><td>0.91</td><td>0.92</td><td>1272524</td></tr> <tr> <td>weighted avg</td><td>1.00</td><td>1.00</td><td>1.00</td><td>1272524</td></tr> </tbody> </table> <p>Accuracy: 0.9996</p> <p><b>Confusion Matrix:</b></p> <table> <tr> <td>[[1270669 212]</td> </tr> <tr> <td>[ 284 1359]]</td> </tr> </table>		precision	recall	f1-score	support	0	1.00	1.00	1.00	1270881	1	0.87	0.83	0.85	1643	accuracy			1.00	1272524	macro avg	0.93	0.91	0.92	1272524	weighted avg	1.00	1.00	1.00	1272524	[[1270669 212]	[ 284 1359]]
	precision	recall	f1-score	support																													
0	1.00	1.00	1.00	1270881																													
1	0.87	0.83	0.85	1643																													
accuracy			1.00	1272524																													
macro avg	0.93	0.91	0.92	1272524																													
weighted avg	1.00	1.00	1.00	1272524																													
[[1270669 212]																																	
[ 284 1359]]																																	
KNN	<p><b>Optimized KNN Evaluation:</b></p> <p><b>Classification Report:</b></p> <table> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>1.00</td><td>1.00</td><td>1.00</td><td>1270881</td></tr> <tr> <td>1</td><td>0.96</td><td>0.60</td><td>0.74</td><td>1643</td></tr> <tr> <td>accuracy</td><td></td><td></td><td>1.00</td><td>1272524</td></tr> <tr> <td>macro avg</td><td>0.98</td><td>0.80</td><td>0.87</td><td>1272524</td></tr> <tr> <td>weighted avg</td><td>1.00</td><td>1.00</td><td>1.00</td><td>1272524</td></tr> </tbody> </table> <p>Accuracy: 0.9995</p> <p><b>Confusion Matrix:</b></p> <table> <tr> <td>[[1270838 43]</td> </tr> <tr> <td>[ 651 992]]</td> </tr> </table>		precision	recall	f1-score	support	0	1.00	1.00	1.00	1270881	1	0.96	0.60	0.74	1643	accuracy			1.00	1272524	macro avg	0.98	0.80	0.87	1272524	weighted avg	1.00	1.00	1.00	1272524	[[1270838 43]	[ 651 992]]
	precision	recall	f1-score	support																													
0	1.00	1.00	1.00	1270881																													
1	0.96	0.60	0.74	1643																													
accuracy			1.00	1272524																													
macro avg	0.98	0.80	0.87	1272524																													
weighted avg	1.00	1.00	1.00	1272524																													
[[1270838 43]																																	
[ 651 992]]																																	

	<b>Optimized Gradient Boosting Evaluation:</b> <b>Classification Report:</b> <table border="1"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>1.00</td><td>1.00</td><td>1.00</td><td>1270881</td></tr> <tr> <td>1</td><td>0.84</td><td>0.61</td><td>0.71</td><td>1643</td></tr> <tr> <td><b>accuracy</b></td><td></td><td></td><td><b>1.00</b></td><td><b>1272524</b></td></tr> <tr> <td><b>macro avg</b></td><td><b>0.92</b></td><td><b>0.81</b></td><td><b>0.85</b></td><td><b>1272524</b></td></tr> <tr> <td><b>weighted avg</b></td><td><b>1.00</b></td><td><b>1.00</b></td><td><b>1.00</b></td><td><b>1272524</b></td></tr> </tbody> </table> <b>Accuracy: 0.9993</b> <b>Confusion Matrix:</b> $[[1270684 \quad 197] \\ [ \quad 633 \quad 1010]]$		precision	recall	f1-score	support	0	1.00	1.00	1.00	1270881	1	0.84	0.61	0.71	1643	<b>accuracy</b>			<b>1.00</b>	<b>1272524</b>	<b>macro avg</b>	<b>0.92</b>	<b>0.81</b>	<b>0.85</b>	<b>1272524</b>	<b>weighted avg</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1272524</b>
	precision	recall	f1-score	support																											
0	1.00	1.00	1.00	1270881																											
1	0.84	0.61	0.71	1643																											
<b>accuracy</b>			<b>1.00</b>	<b>1272524</b>																											
<b>macro avg</b>	<b>0.92</b>	<b>0.81</b>	<b>0.85</b>	<b>1272524</b>																											
<b>weighted avg</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1272524</b>																											

### Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest	Random Forest emerged as the recommended model. It provided the best balance between high fraud precision (minimizing false positives) and reasonable recall (catching most frauds), which is crucial for business operations. Gradient Boosting and Decision Tree models either produced excessive false positives or missed too many frauds, while KNN struggled with the high dimensionality and imbalance of the data.