From Network Traffic Measurements to QoE for Internet Video

ARBAI Safaa
MADEIRA DE CARVALHO Martim
LE Phuc Lai

Table of Contents

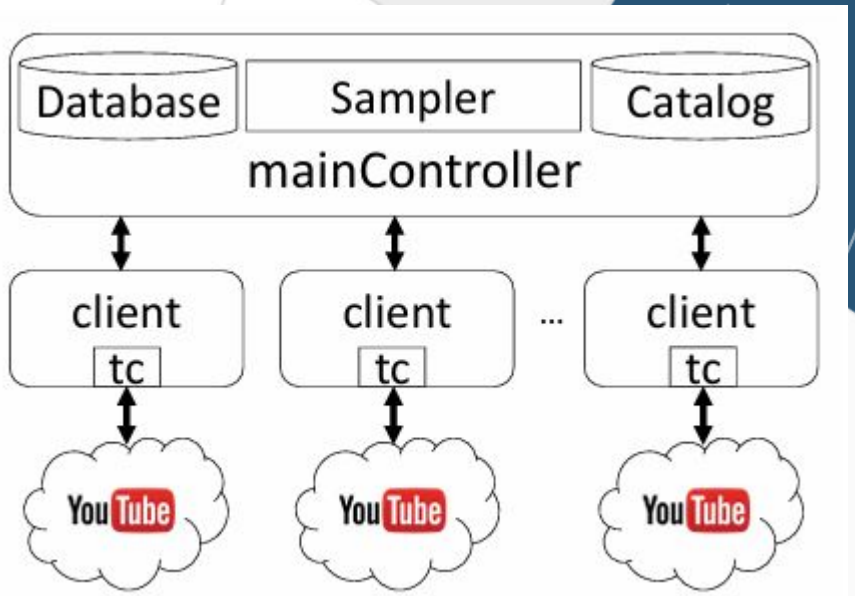
Purpose

- Video Traffic are encrypted (HTTP, QUIC).
- Network operator can only have access to Quality of Service (QoS) signals like throughput, packet sizes, loss, jitter, etc.

- **Goals:** Can we predict user-perceived Quality of Experience (QoE), and other QoS metrics that affect QoE (stall, join_time, etc.) from network and packet level data?

The Dataset

- Trace-based sampling: use real life network trace (RTR-NetzTest, MobiPerf)
- Testbed: Grid5000 and R2Lab platforms, controller on EC2
- Data Collection Process: Client enforce QoS condition using *tc*, video is played on Chrome browse
- Feature extraction:
  - Network Layer: Statistical features were gathered from encrypted packet traces: throughput, interarrival time
  - Application Layer: HTTP traces and the YouTube API were used to record "ground truth" QoE metrics: startup delay, stalls, resolution changes.
- Ground Truth Labeling (Subjective MOS): ITU-T P.1203



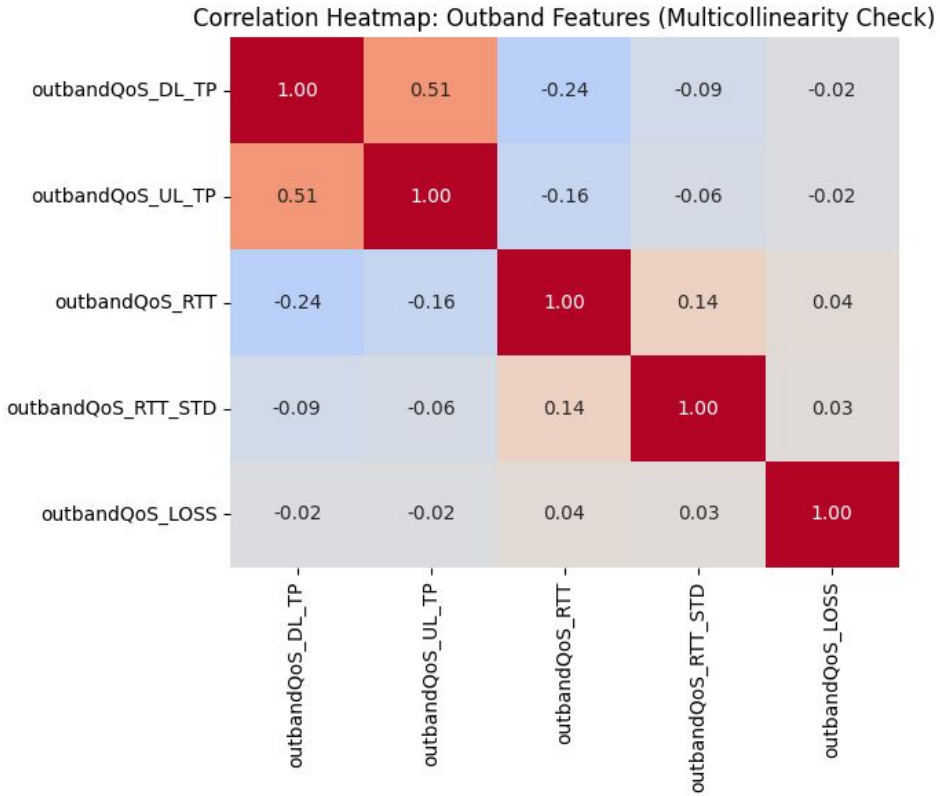
The experimentation framework

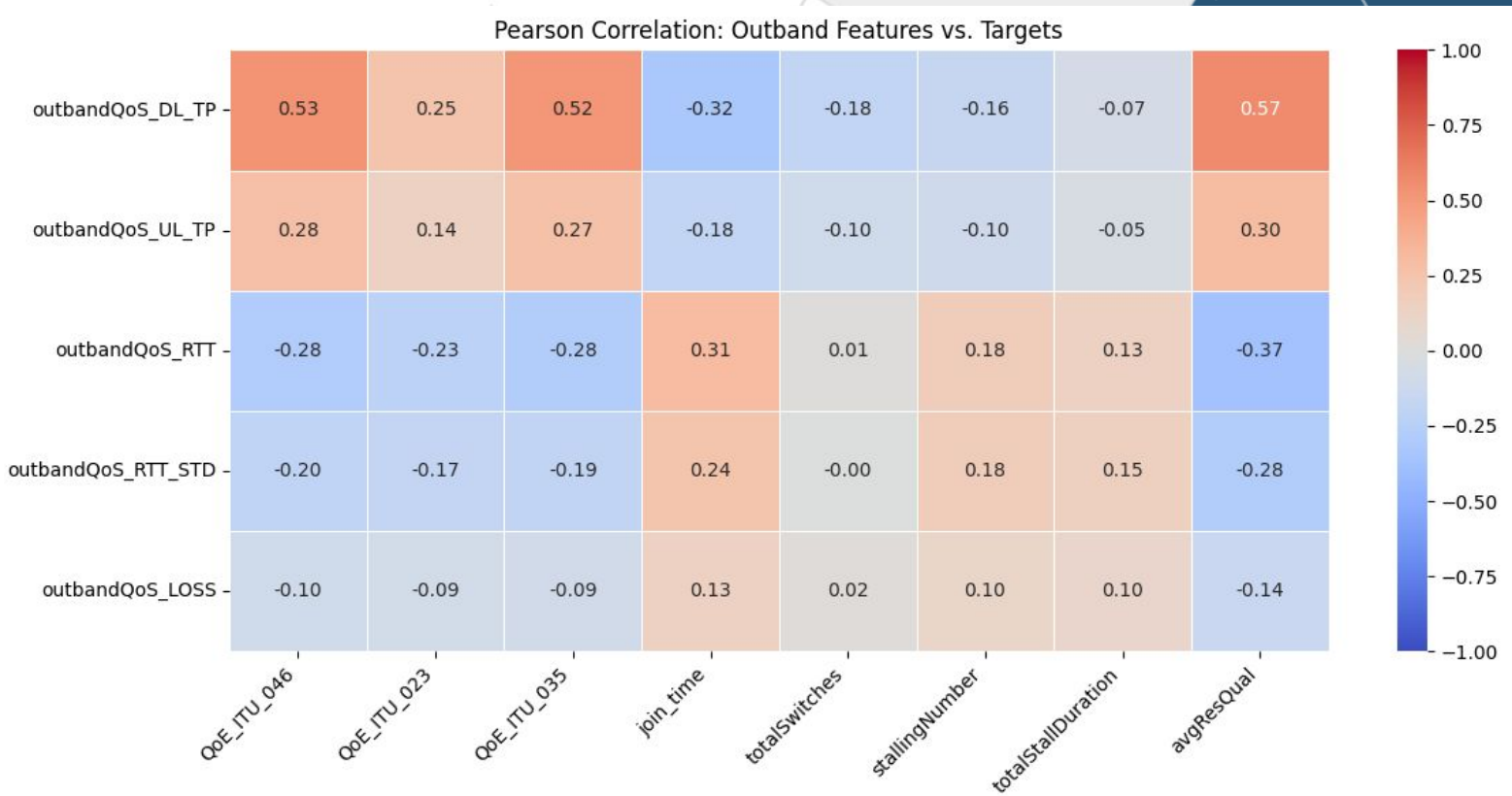
From Network Traffic Measurements to QoE for Internet Video | IEEE Conference Publication | IEEE Xplore

The Dataset

- **F_inband:** 48 features
  - Encrypted packet traces, include the statistical metrics of the downlink throughput (bps), the uplink and downlink interarrival times (s) and the downlink packet sizes (bytes).
- **F_chunks:** 7 features
  - Chunk information from encrypted traffic. Composed of the the statistical metrics of the chunk sizes array
- **F_outband:** 5 features
  - The feature that are configured on *tc*
- **Target:** ITU_QoE, join time, average resolution, stalling, startup delay, quality switch
- **Other feature:** Video category

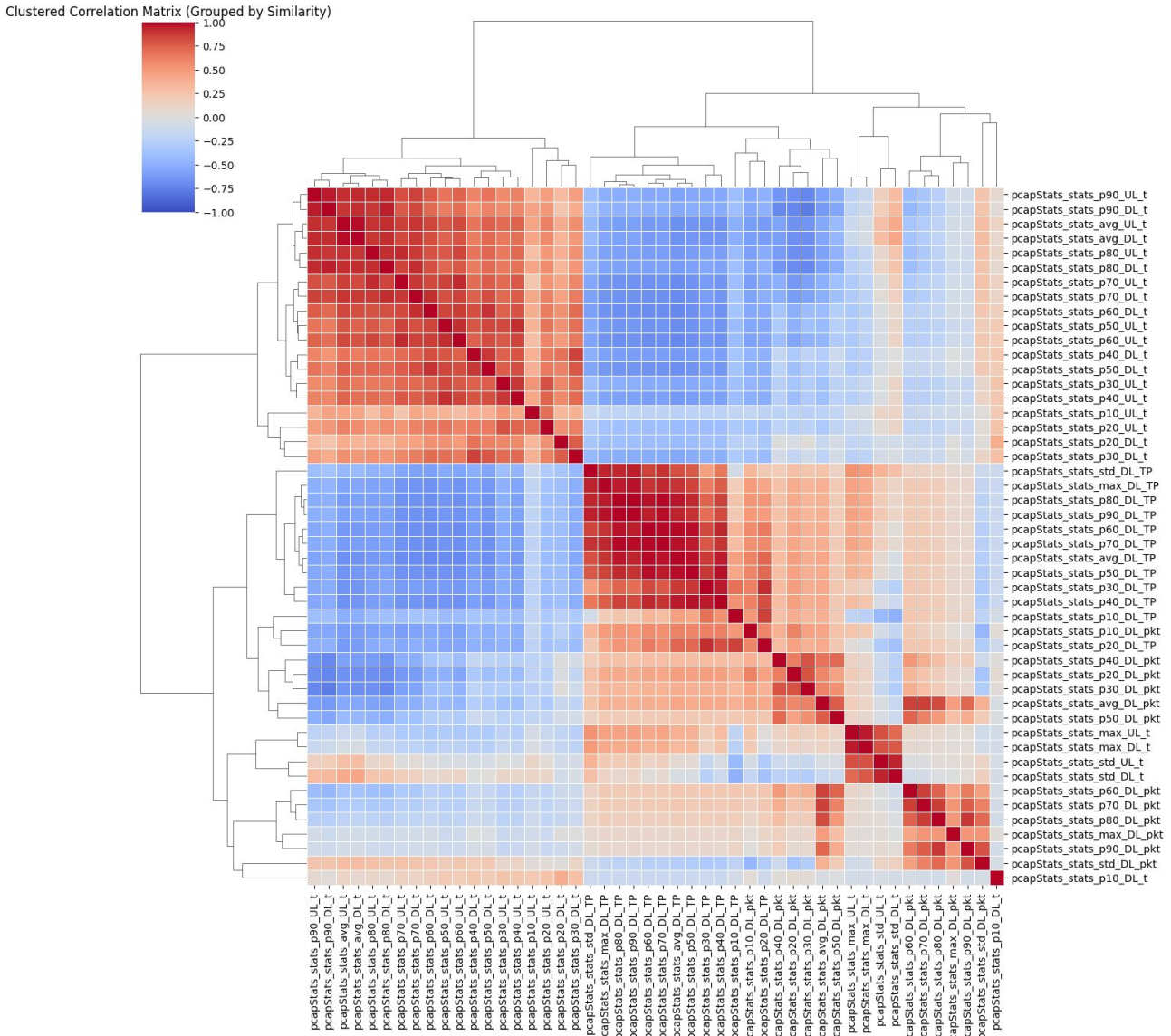
Data Exploration





Data Exploration

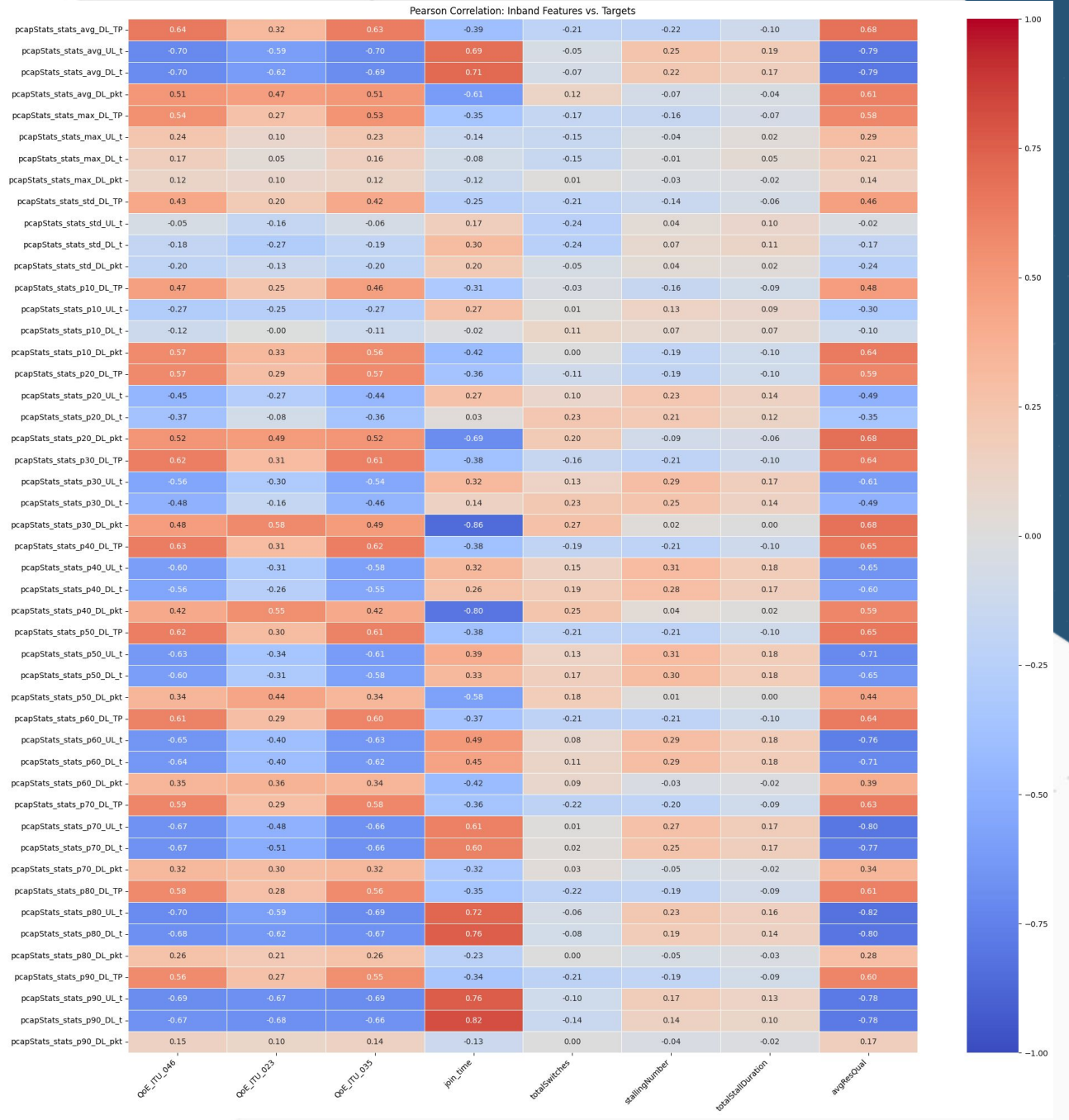
Lots of feature have high collinear in **F_inband**



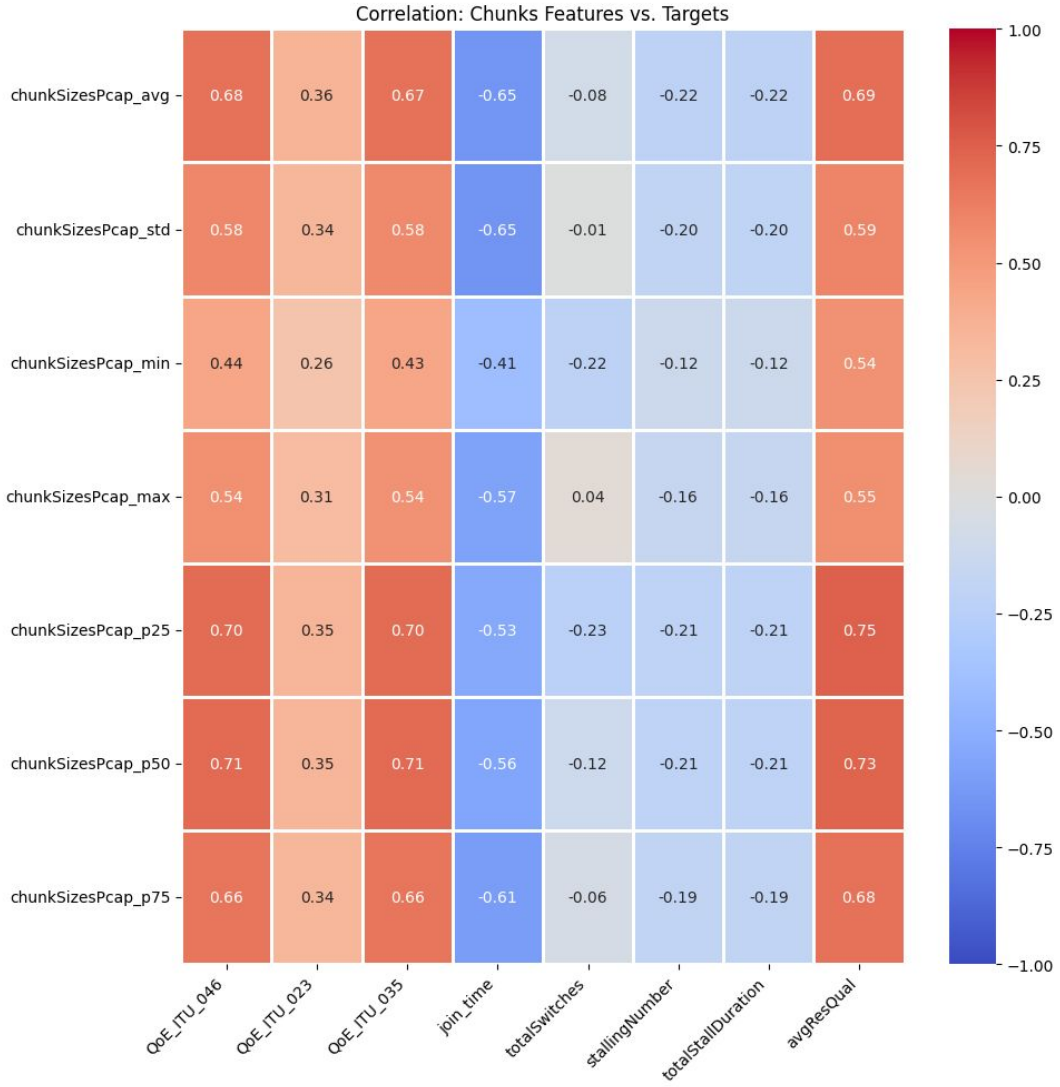
- ***_UL_t** and ***_DL_t** (round trip time) features are heavily correlated
- ***_DL_TP** (throughput) and ***_DL_pkt** (packet size) negative correlated to ***_UL_t** and ***_DL_t**
- Form **p30_DL_TP** and up, these feature are effectively clones of each other => Redundancy?
- The same applies for **p60_DL_pkt** and up, for **std_UL_t** and **max_UL_t**

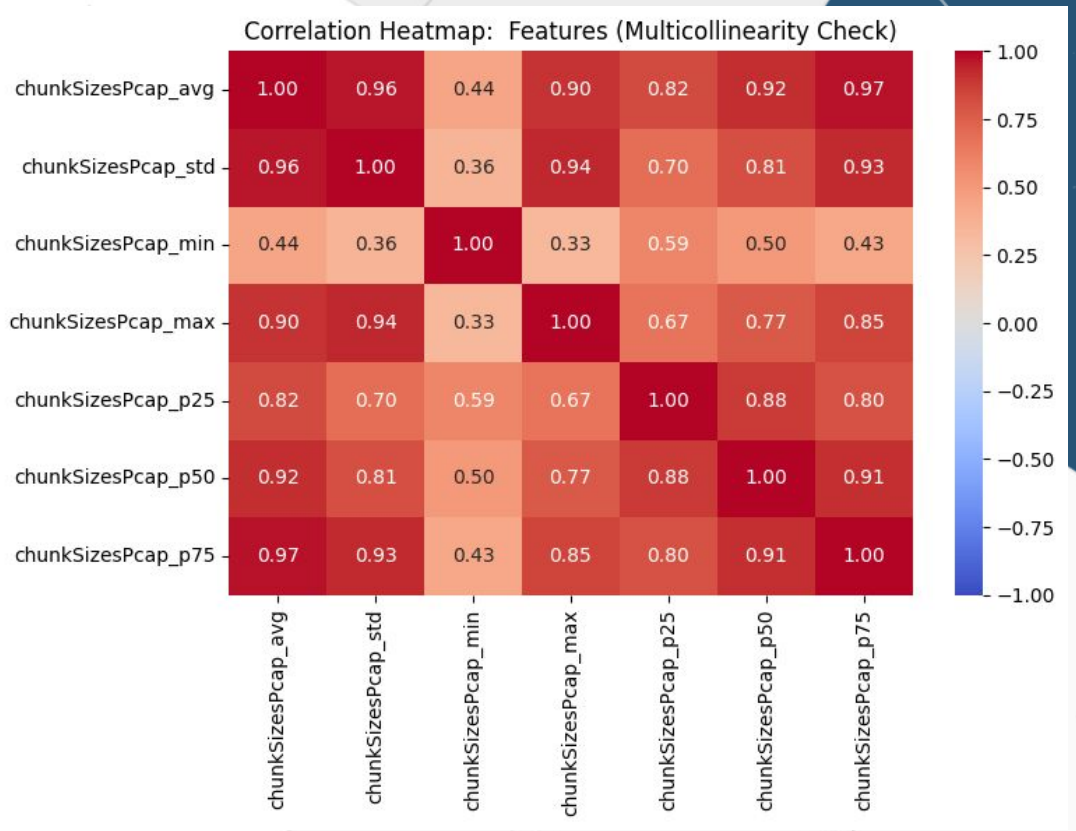
Data Exploration

In **F_inband**, round trip time of both upload and download affect the join time while packet size, download throughput affect the QoE and average resolution

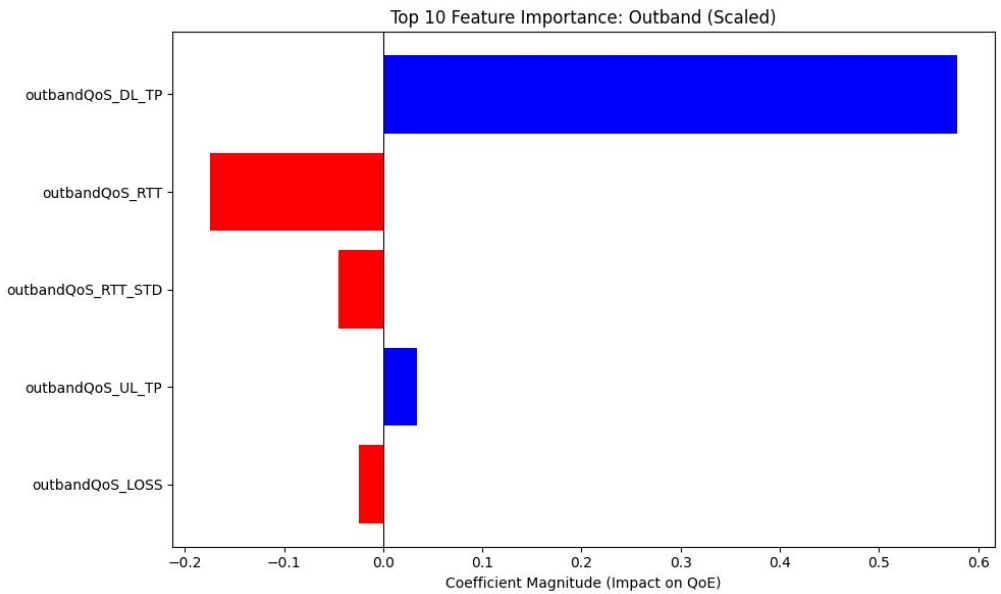


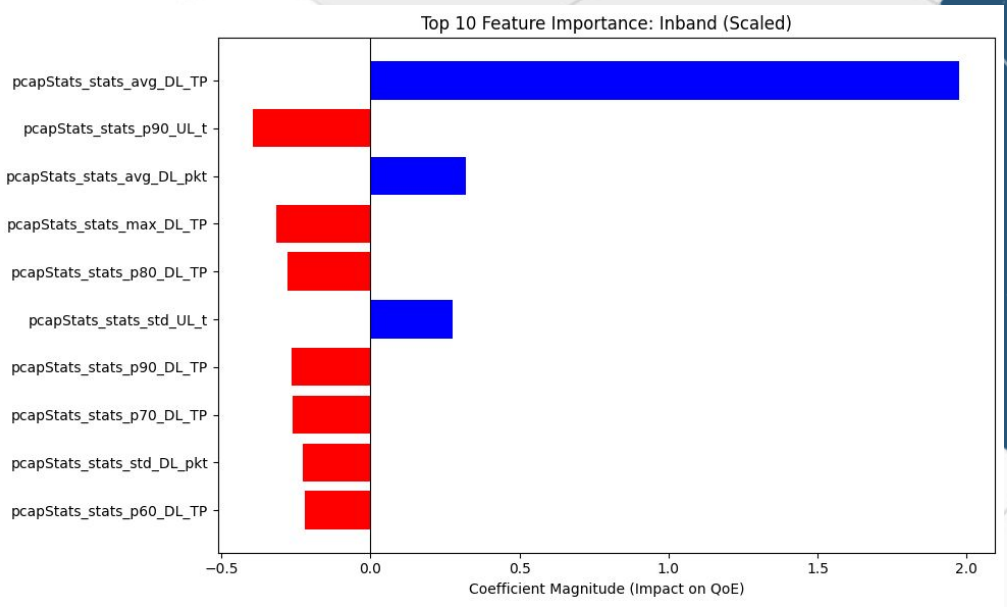
Data Exploration





Linear Regression with Scaled Data



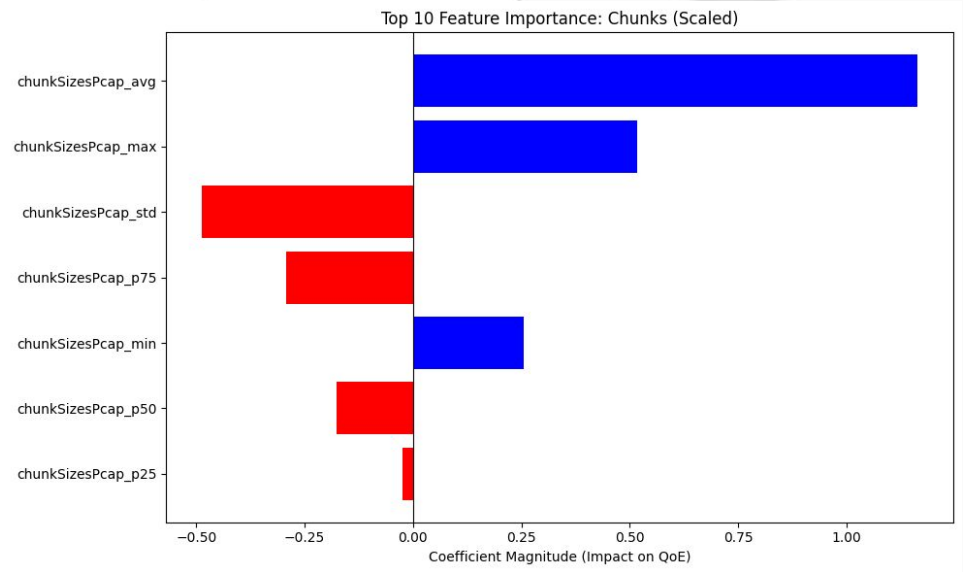


Non-Scaled MSE:

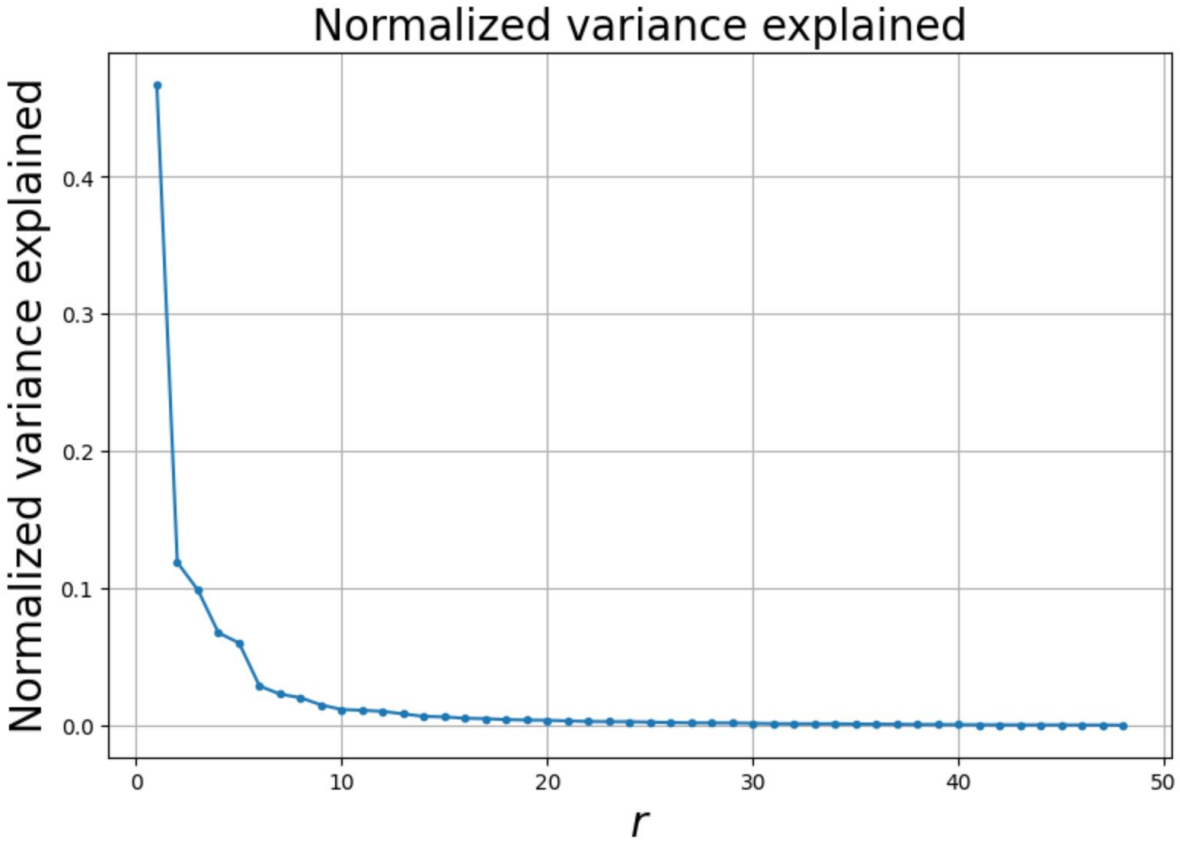
- Outband : 1.04
- Inband : 0.57
- Chunks : 0.79

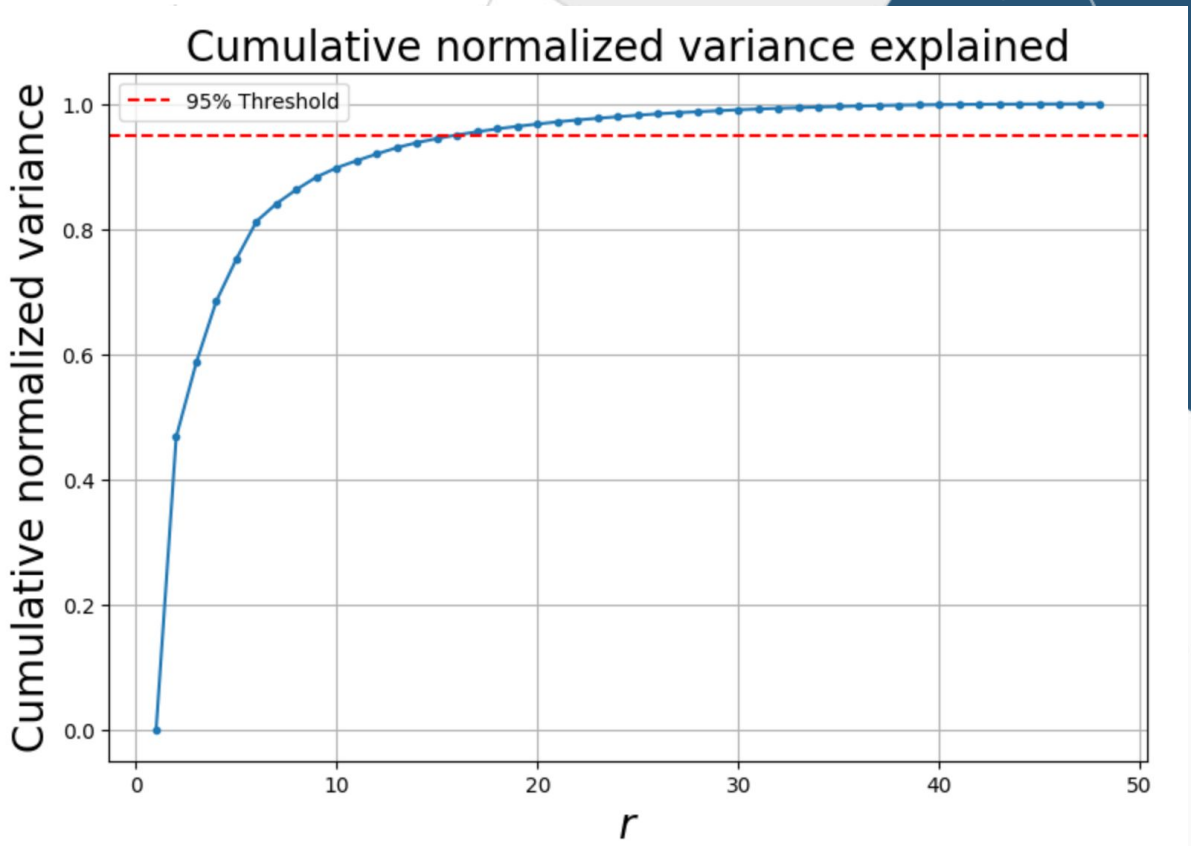
Scaled MSE:

- Outband : 1.04
- Inband : 0.57
- Chunks : 0.78



Dimensionality Reduction

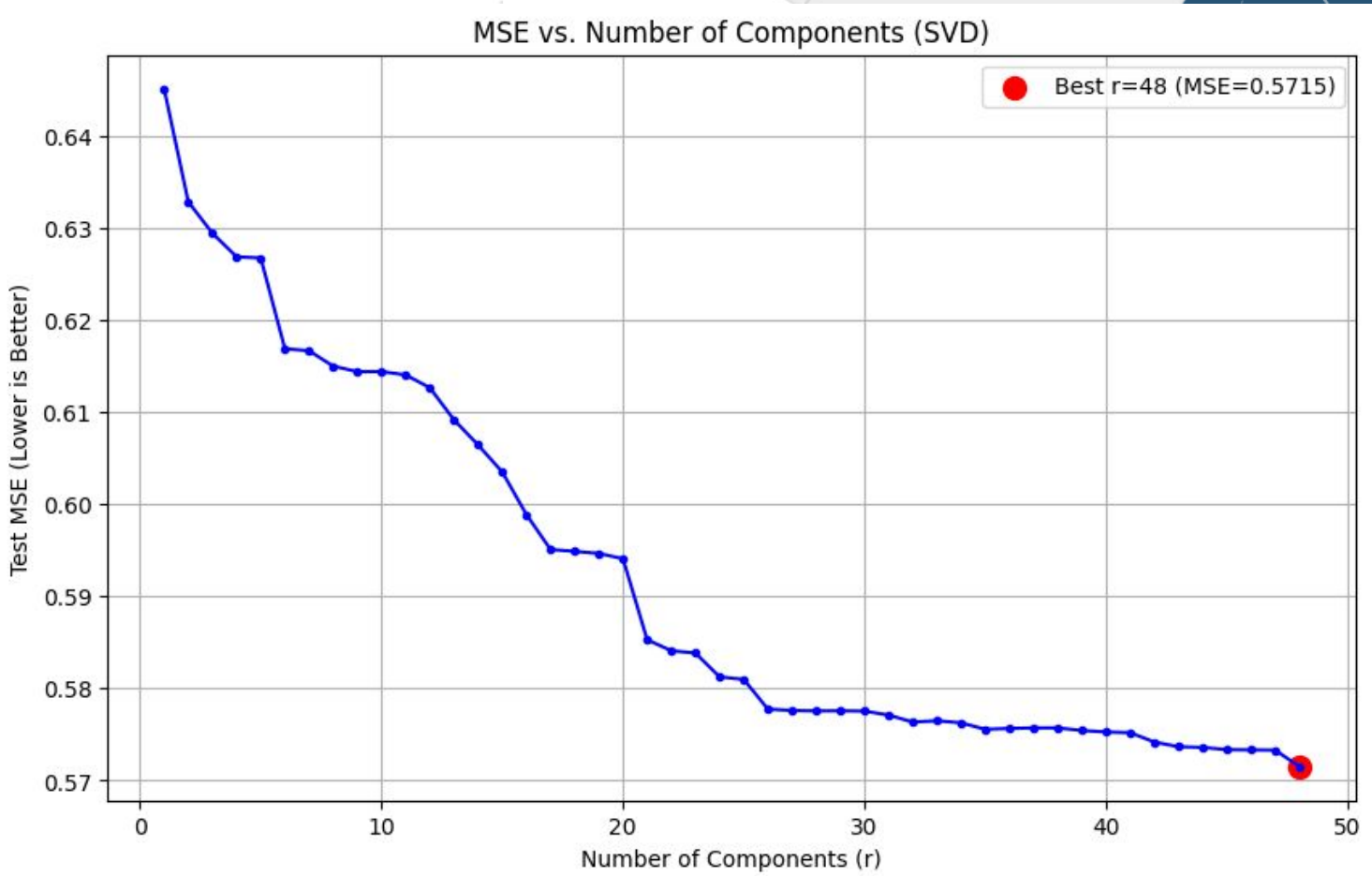




Dimensionality Reduction

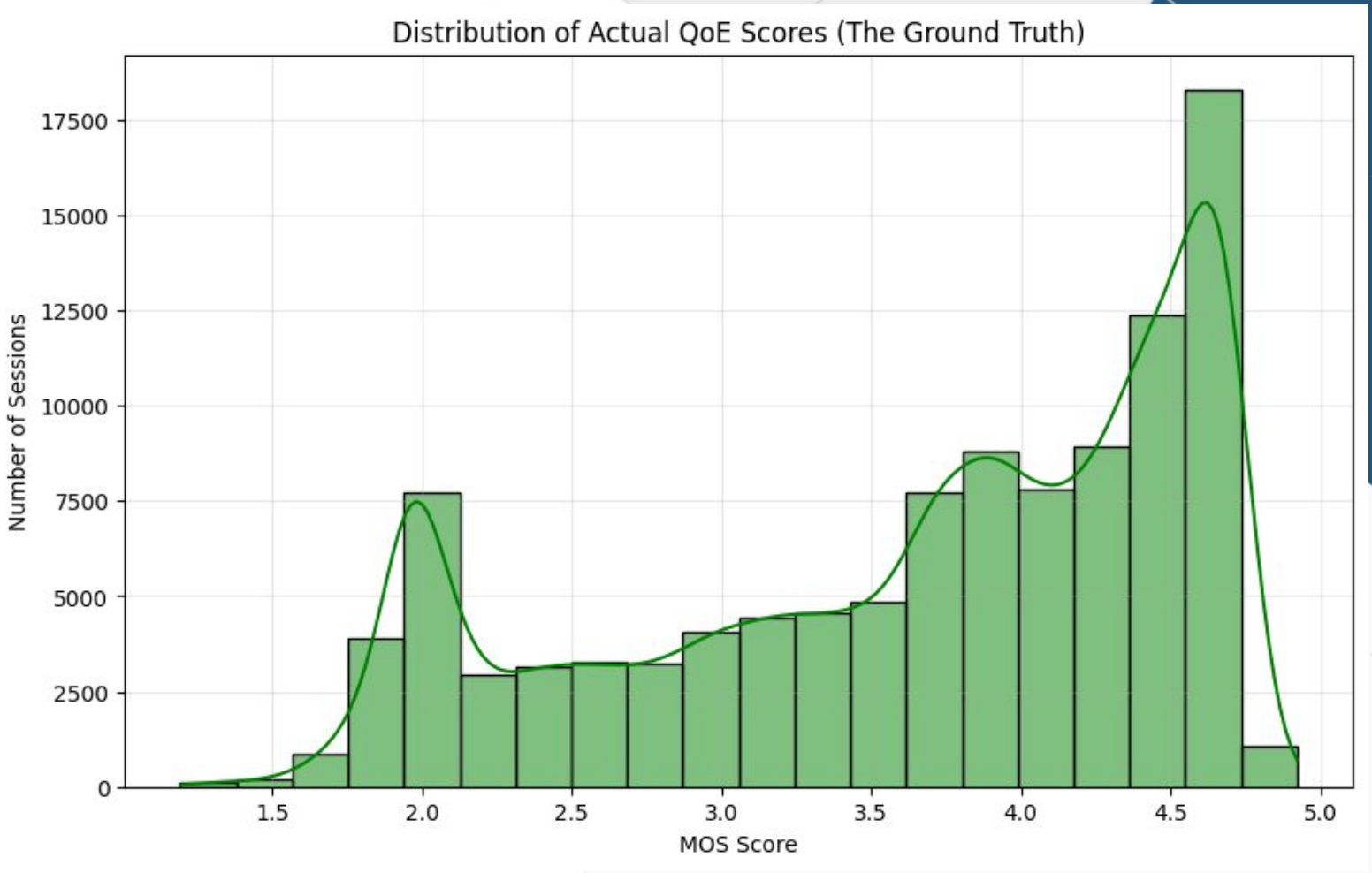
- F_inband full 48 features:
  - MSE: 5.5715
- F_inband reduced to 30 features
  - MSE: 0.5775

Dimensionality Reduction is not useful in Linear Regression for F_inband features



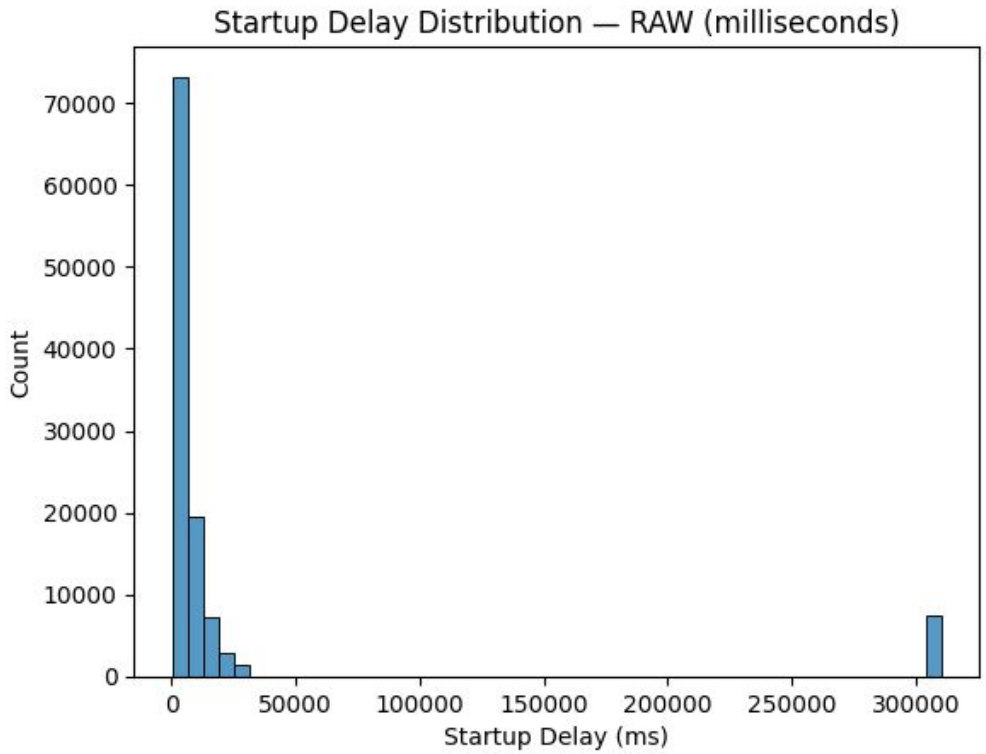
Data Exploration

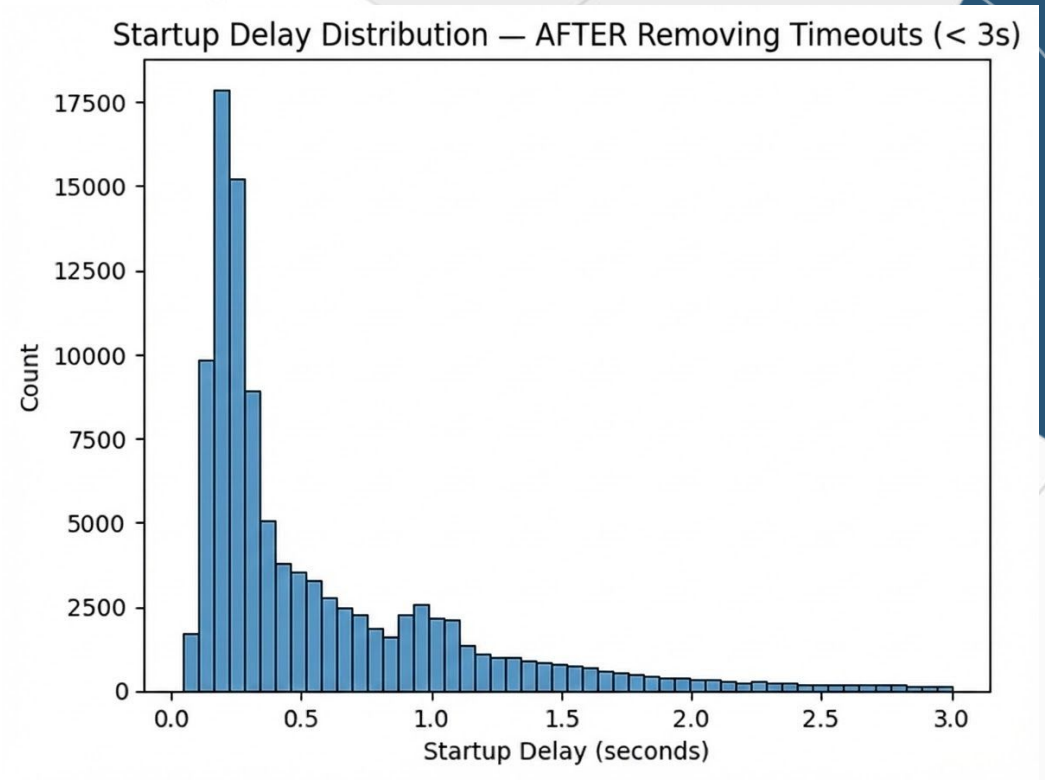
- Most of the ground truth QoE data are good QoE (4-5)
- Average QoE Score: 3.6



Data Exploration

Startup Delay

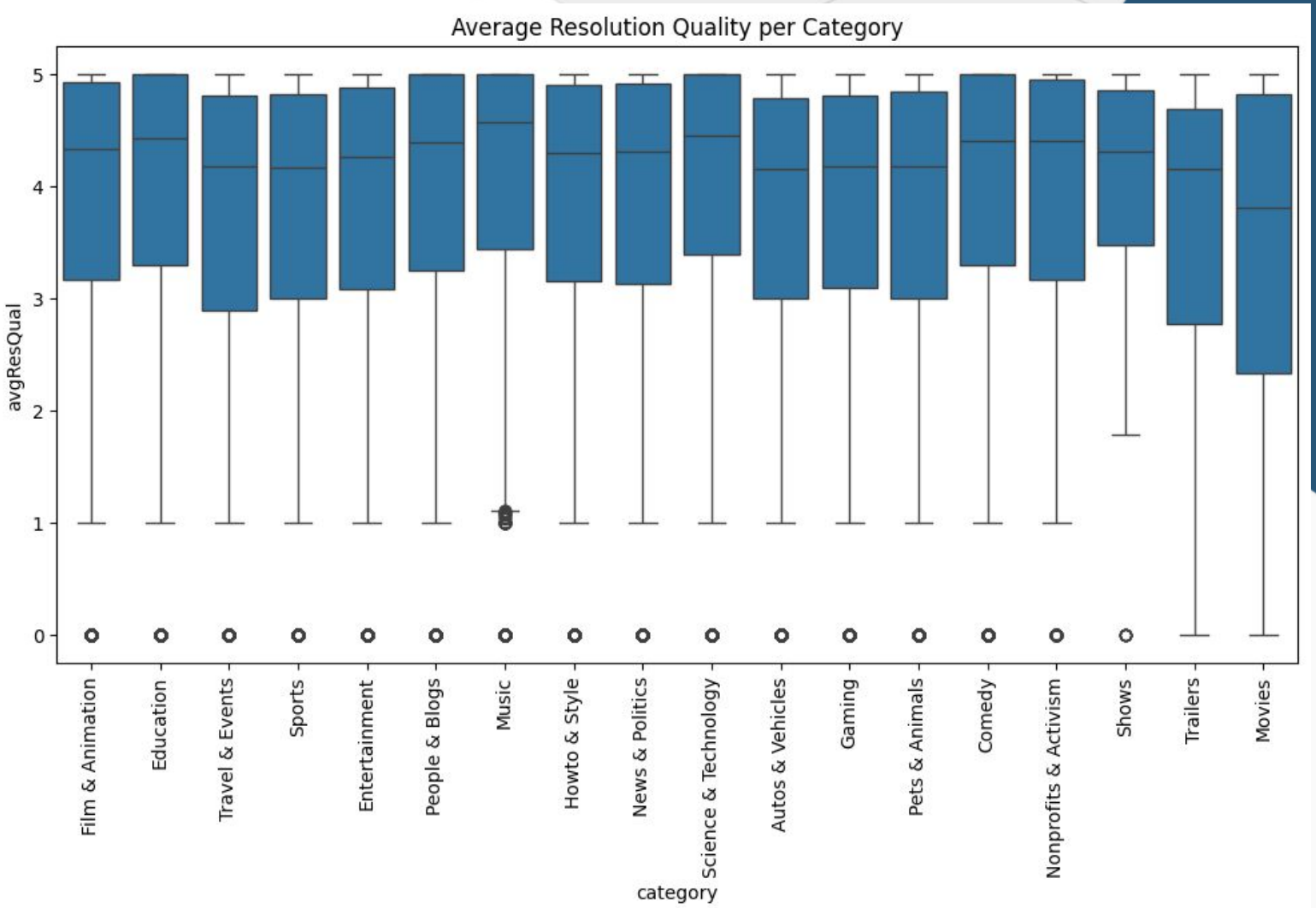




Key Takeaway: Timeouts are measurement artifacts and must be removed to model true startup delay.

Data Exploration

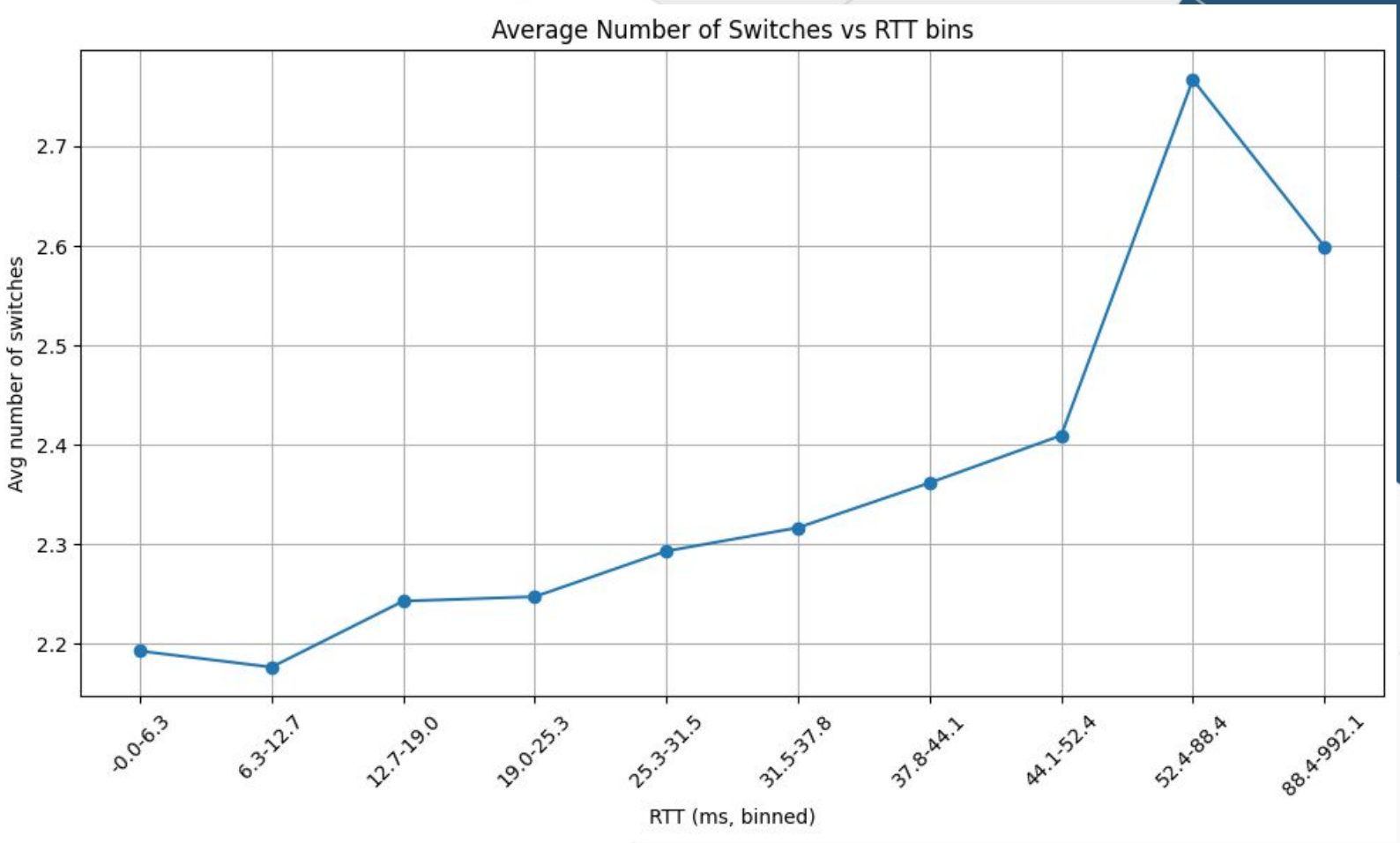
Category of video does not affect the resolution



Data Exploration

Bin 52.4-88.4 is interesting:

- Youtube specs
- “midpoint” → too bad for high quality / too good for low quality



Feature Visibility Regimes - Definition

Three Feature Visibility Regimes

Minimal (Out-of-band QoS)

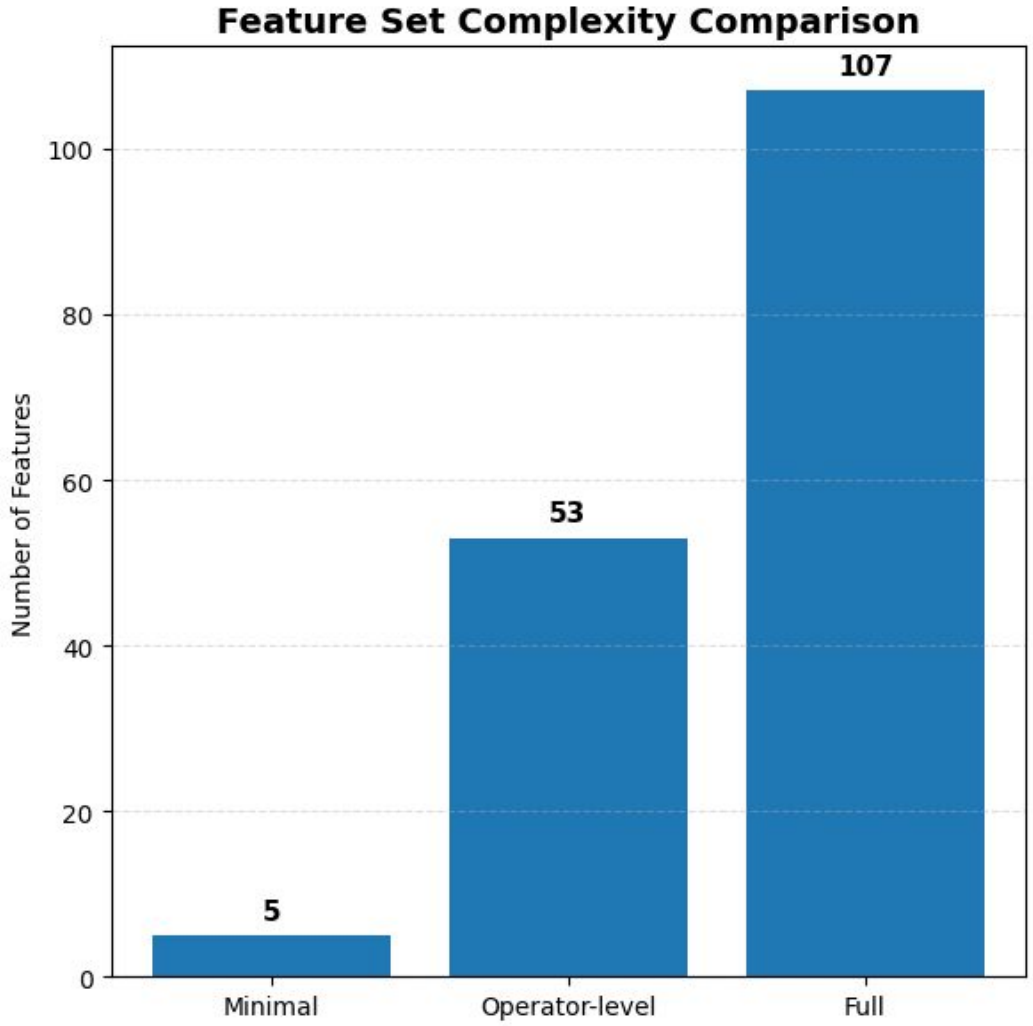
- Network parameters measurable without deep packet inspection
- RTT, bandwidth (DL/UL), packet loss; available to all operators
- Ground truth QoS: Known in real life up to an error margin

Operator (Minimal + Encrypted Packet Statistics)

- Adds packet-level metrics from encrypted traffic (pcapStats)
- Observable despite encryption (timing, sizes, patterns)

Full (Operator + Inferred Chunk-level Features)

- Requires advanced inference of application-layer metrics
- Chunk sizes extracted via clustering



Stall Detection
(Classification / anomaly detection)

Stall Detection

Random Forest:

(balanced db)

	precision	recall	f1-score	support
0	0.90	0.99	0.94	17727
1	0.63	0.14	0.23	2273
accuracy			0.89	20000
macro avg	0.76	0.57	0.59	20000
weighted avg	0.87	0.89	0.86	20000

ROC-AUC: 0.8128296145050398

Feed-forward Neural Network:

(η=0.001; 20 epochs; 2 hidden layers)

	precision	recall	f1-score	support
0	0.96	0.75	0.84	17727
1	0.28	0.76	0.41	2273
accuracy			0.75	20000
macro avg	0.62	0.76	0.63	20000
weighted avg	0.88	0.75	0.79	20000

ROC-AUC: 0.8210555129390565

used features

```python
from sklearn.model_selection import train_test_split
df_s["has_stall"] = (df_s["stallingNumber"] > 0).astype(int)


target = "has_stall"
feature_cols = [
    "outbandQoS_DL_TP",
    "outbandQoS_UL_TP",
    "outbandQoS_RTT",
    "outbandQoS_RTT_STD",
    "outbandQoS_LOSS",
    "pcapStats_stats_avg_DL_TP",
    "pcapStats_stats_std_DL_TP",
    "pcapStats_stats_p90_DL_TP",
    "pcapStats_stats_avg_DL_t",
    "pcapStats_stats_avg_DL_pkt",
]
```

Stall duration:

Random forest regressor (200 trees):

(applied log transformation)

```
RMSE (log-scale): 2.2256833626676964
MAE  (log-scale): 1.2606616503699768
RMSE (Milliseconds): 9589.460973564303
MAE  (Milliseconds): 796.5683536534974
```

Conclusion:

RF → bad recall

NN → bad precision

Stall duration: Not applicable

Stall Detection

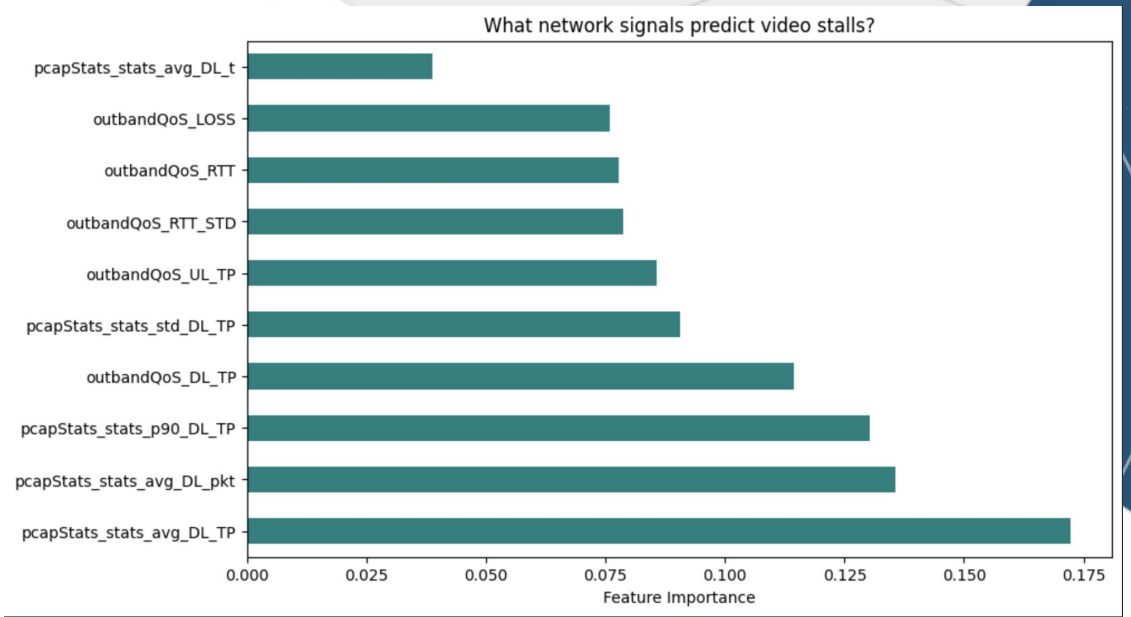
```
Data loaded. Total samples: 100000
Class Balance: {0: 0.88635, 1: 0.11365}

--- Training ISP-View (Network Only) ---
Using 10 features.
Best Decision Threshold: 0.215
              precision    recall  f1-score   support

           0       0.94      0.87      0.90     17727
           1       0.36      0.56      0.44      2273

    accuracy                           0.84     20000
   macro avg       0.65      0.71      0.67     20000
weighted avg       0.87      0.84      0.85     20000

ROC-AUC Score: 0.8129

--- Training Full-View (Network + Player) ---
Using 16 features.
Best Decision Threshold: 0.205
              precision    recall  f1-score   support

           0       0.95      0.86      0.90     17727
           1       0.37      0.61      0.46      2273

    accuracy                           0.84     20000
   macro avg       0.66      0.74      0.68     20000
weighted avg       0.88      0.84      0.85     20000

ROC-AUC Score: 0.8273
```

Conclusion:

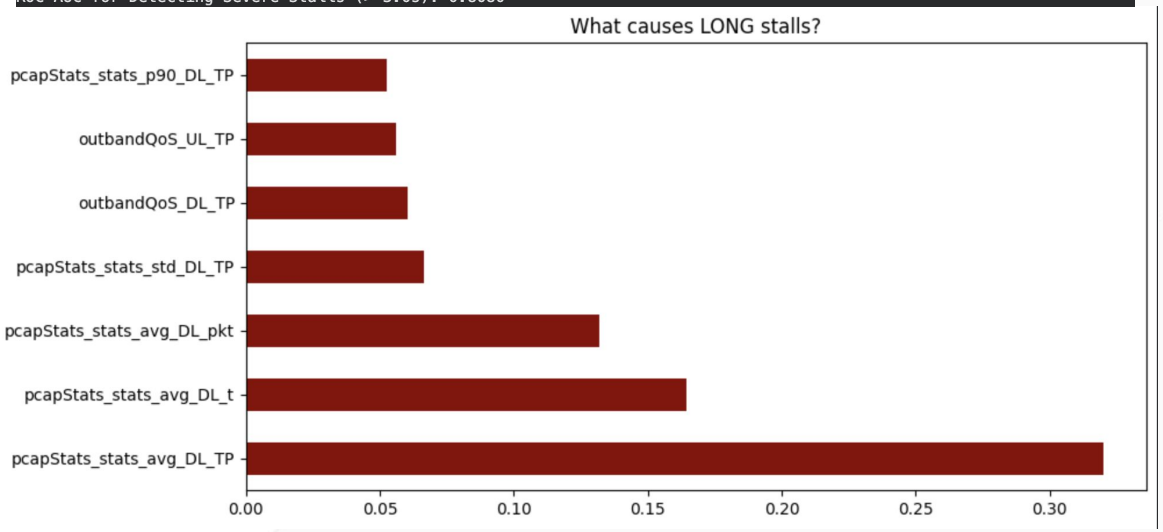
- ISP view ≃ Full view
- Upgrades on model worked
- F1-score is better than before

Stall Detection

```
Data loaded. Total samples: 100000
Class Balance: {0: 0.88635, 1: 0.11365}

--- Training ISP-View (Network Only) ---
Using 10 features.
Best Decision Threshold: 0.215
              precision    recall  f1-score   support

           0       0.94      0.87      0.90     17727
           1       0.36      0.56      0.44      2273

    accuracy                           0.84     20000
   macro avg       0.65      0.71      0.67     20000
weighted avg       0.87      0.84      0.85     20000

ROC-AUC Score: 0.8129

--- Training Full-View (Network + Player) ---
Using 16 features.
Best Decision Threshold: 0.205
              precision    recall  f1-score   support

           0       0.95      0.86      0.90     17727
           1       0.37      0.61      0.46      2273

    accuracy                           0.84     20000
   macro avg       0.66      0.74      0.68     20000
weighted avg       0.88      0.84      0.85     20000

ROC-AUC Score: 0.8273
```

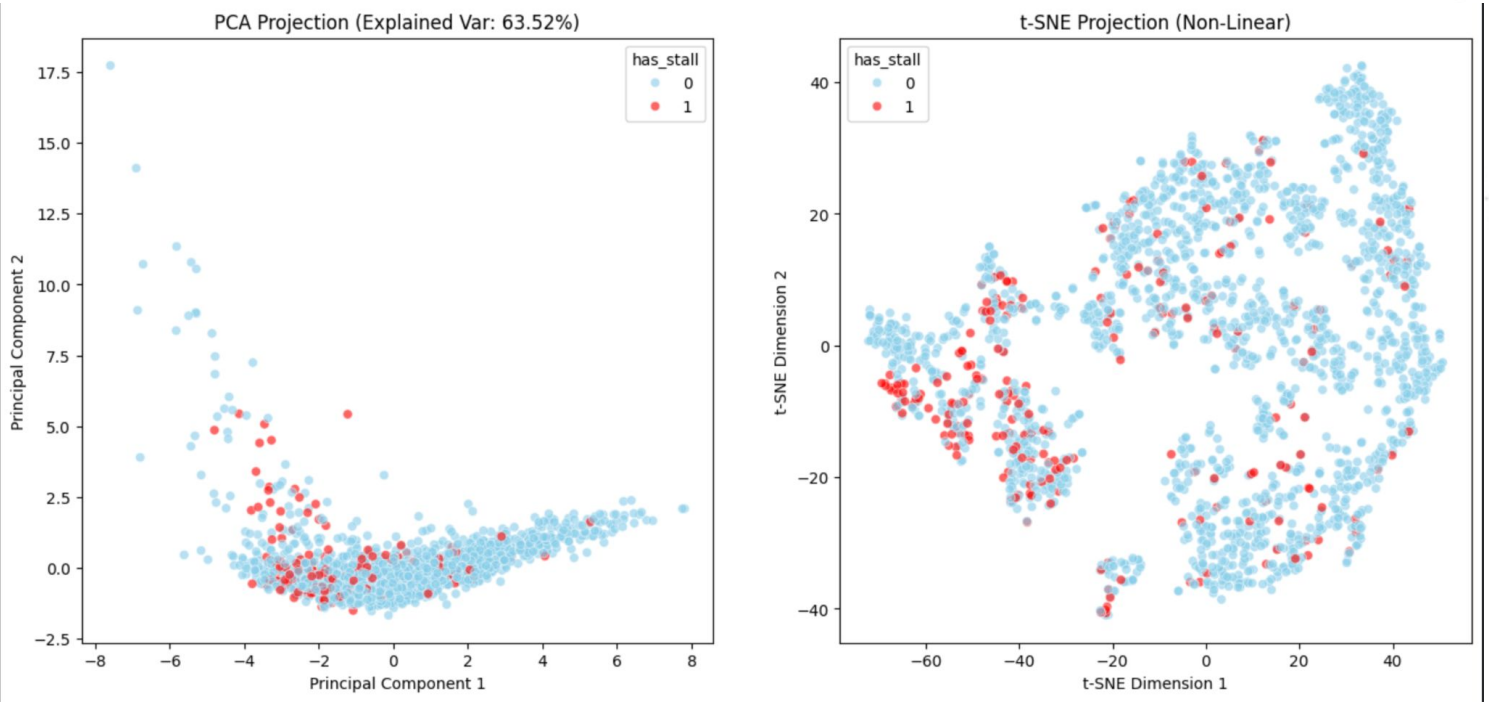



```
Severe Stall Count: {0: 88645, 1: 11355}
ROC-AUC for Detecting Severe Stalls (> 3.0s): 0.8080
```



Stall Detection

Principal Component Analysis vs stochastic neighbor embedding

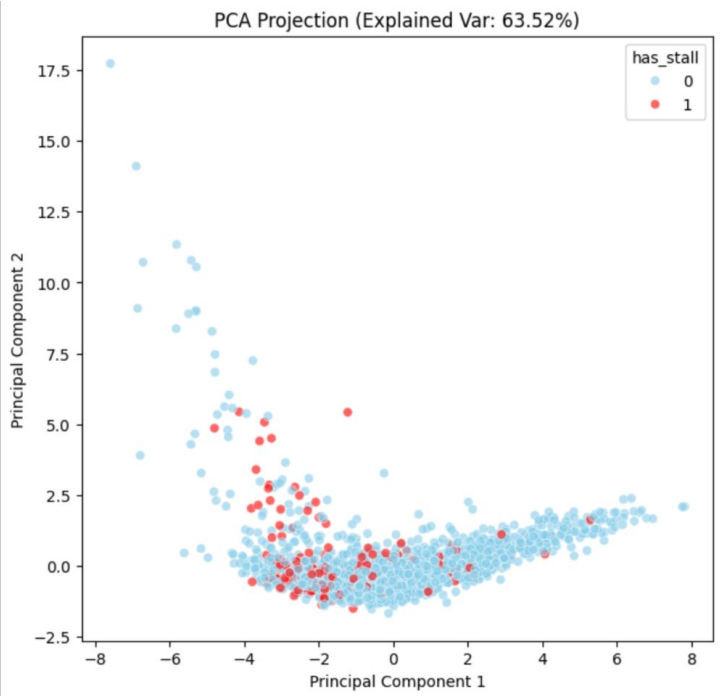


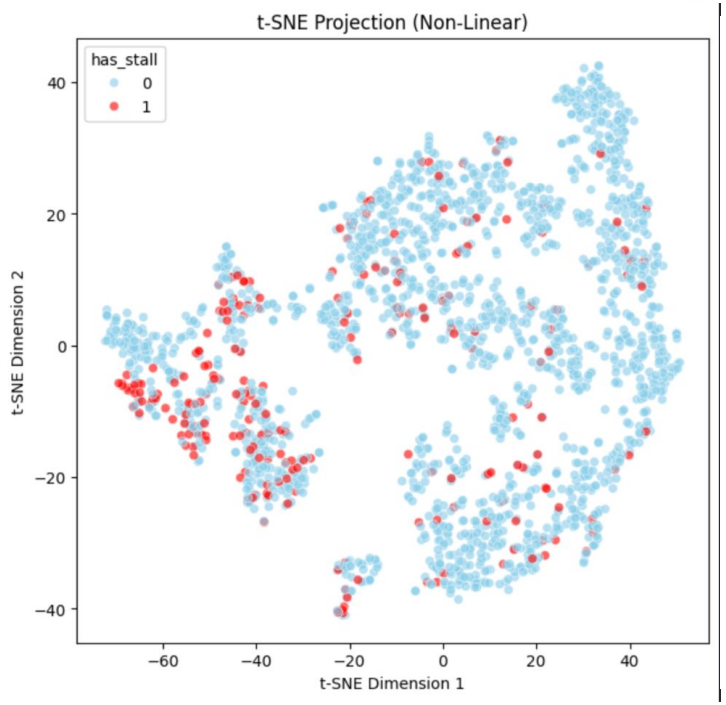
Conclusion:
Points are scattered → buffer health problems ?

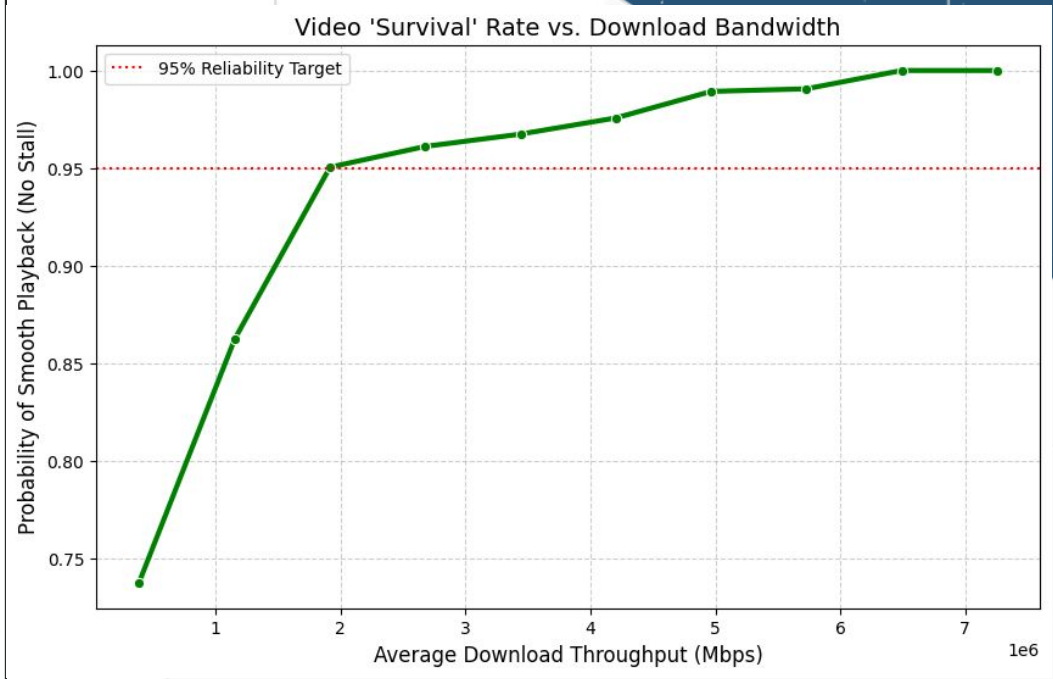
Model can't be a lot more precise than this

Stall Detection

Principal Component Analysis vs stochastic neighbor embedding







Startup Delay Classification (Timeout Detection)

Startup Delay Classification (Timeout Detection) - Motivation

Problem Definition

Objective:

- Detect whether a video session **successfully starts** or **times out** (never started)

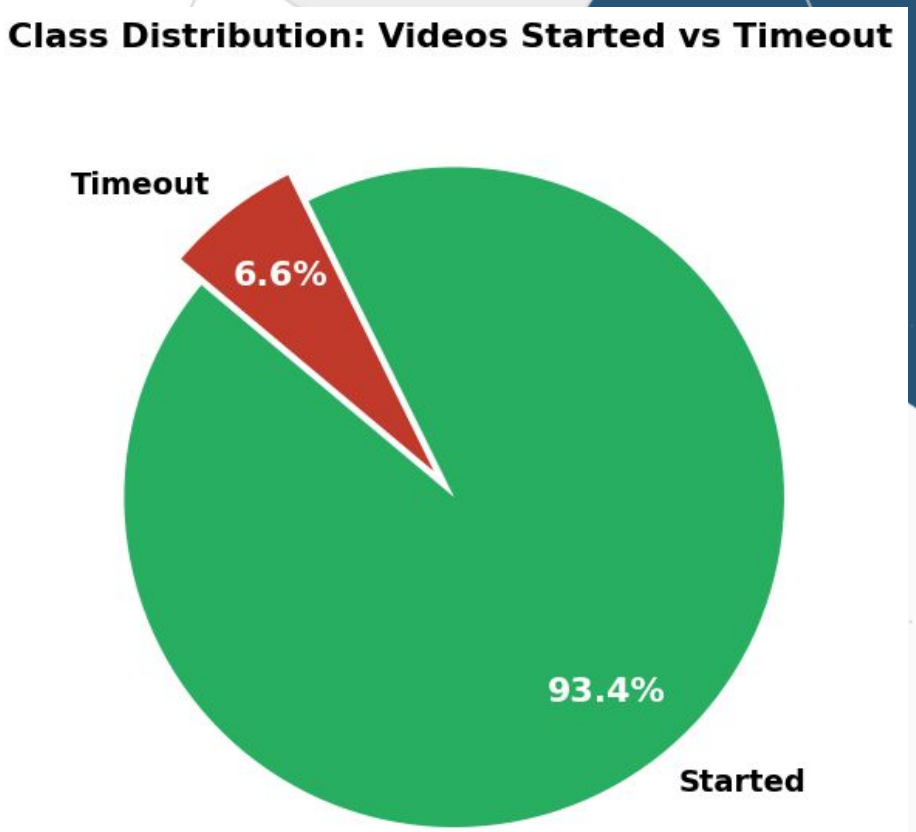
The Critical Problem: Startup Delay

User Impact:

- Users abandon videos if startup exceeds **2 seconds** (Krishnan & Sitaraman, 2012)
- **Each +1s delay → +5.8% abandonment increase** (same study)
- First impression is everything (no second chance)

Expected Impact

Early detection (5-10s, not 30s) → **Proactive intervention** → **Reduced churn**



Random Forest Classifier & Cross-Validation Strategy

Why Random Forest for This Problem?

Advantages:

- **Handles non-linear relationships**
- **Robust to class imbalance**
- **Feature importance**

Hyperparameters Selected:

```
RandomForestClassifier( n_estimators=50,          # 50 trees for ensemble
max_depth=15,             # Limit tree depth to prevent overfitting
min_samples_leaf=20,      # Require 20 samples per leaf (statistical significance)
class_weight='balanced',  # Automatic weighting: n_samples / (n_classes * n_class_samples)
random_state=42           # Reproducibility)
```

Handling Class Imbalance:

1. Stratified Sampling

- Maintains 93.4% / 6.6% ratio in train/test splits
- Prevents model from never seeing minority class in validation

2. Class Weighting (`class_weight='balanced'`)

- Automatically adjusts loss function
- Equivalent to inverse frequency weighting

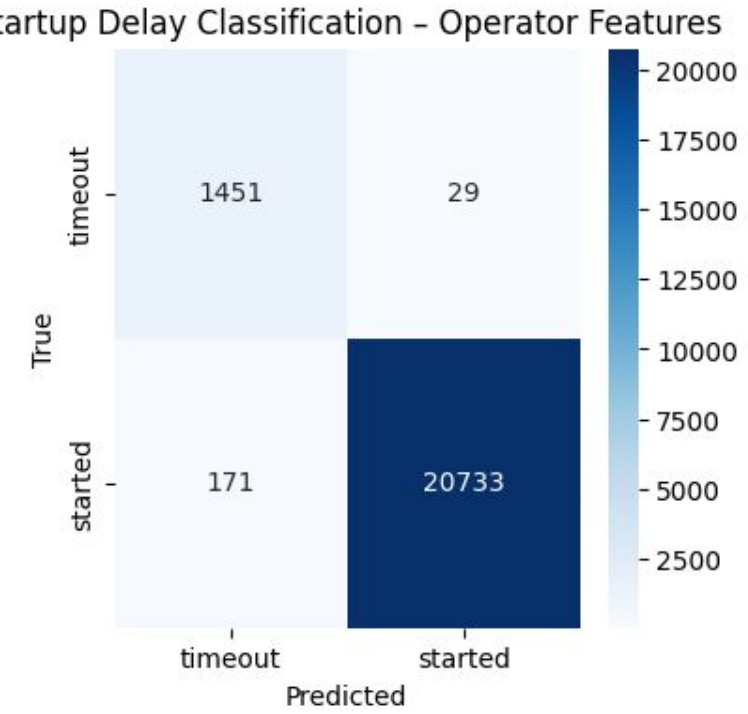
Cross-Validation Strategy (5-Fold Stratified Cross-Validation)

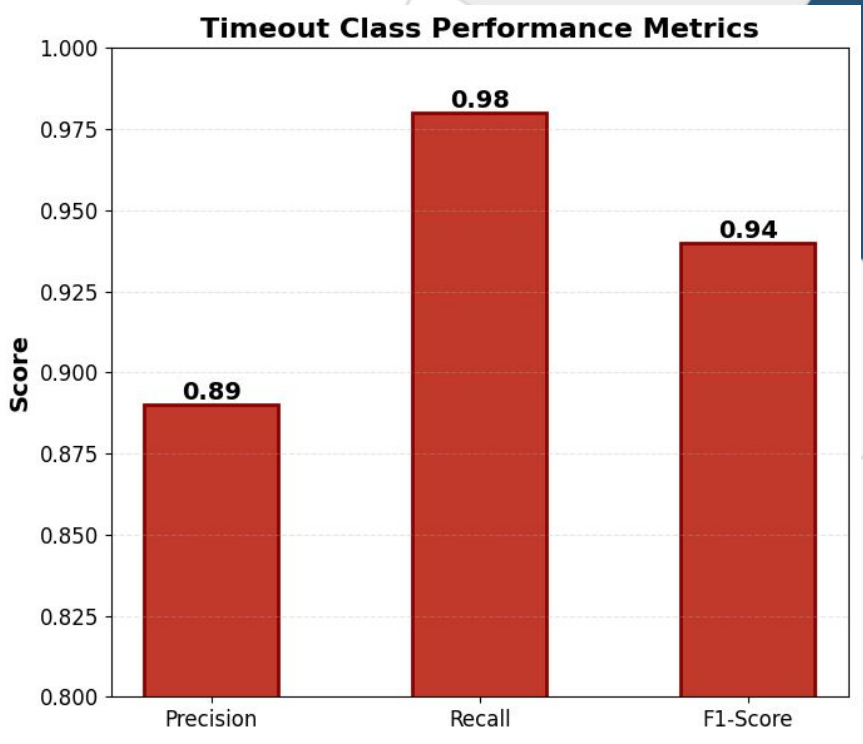
Why Stratified?

- Each fold maintains the 93.4% / 6.6% class distribution
- Critical for imbalanced data (regular K-fold could create folds with few/no timeouts)
- Ensures every fold is representative of the full dataset

Feature Visibility Regimes - Operator Features Results

Classification Report





Key Insight: *Operator-level network features are sufficient for near-perfect timeout detection.*

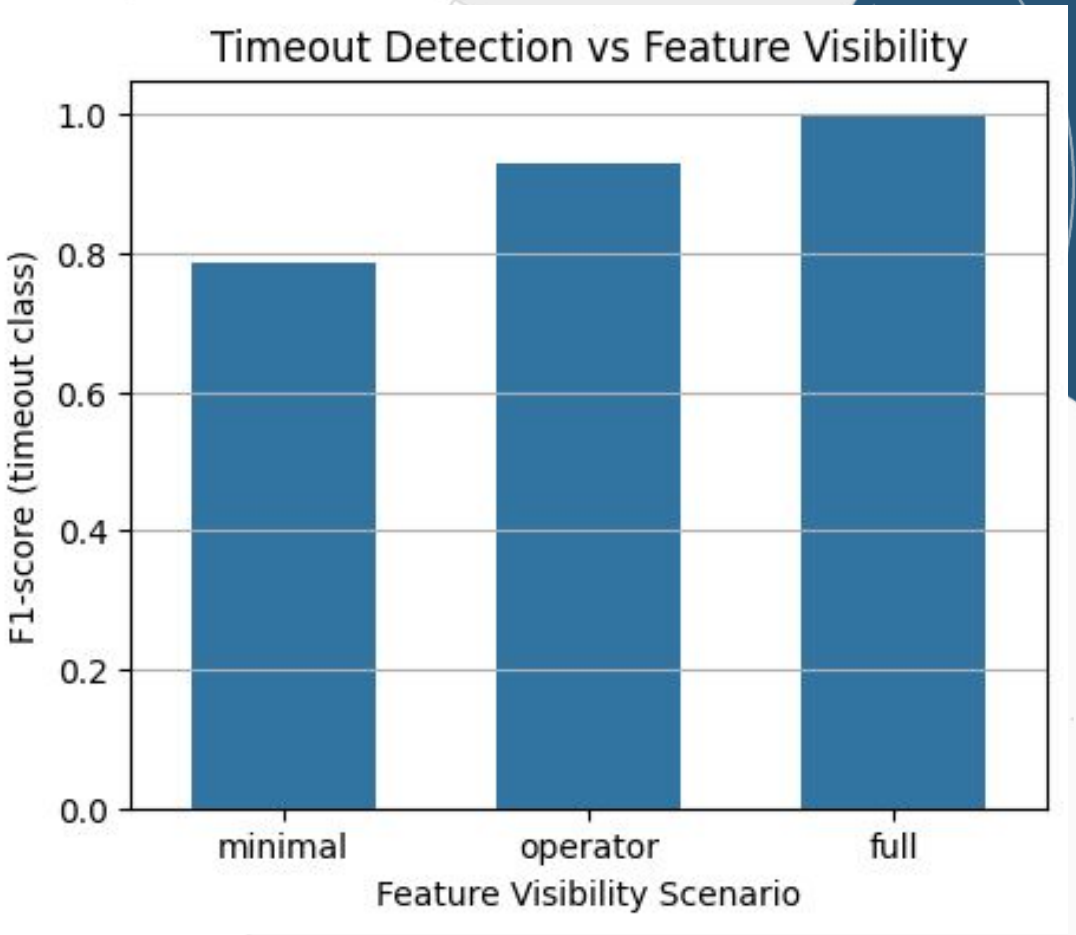
Feature Visibility Comparison - Timeout Detection

Performance Across Feature Sets

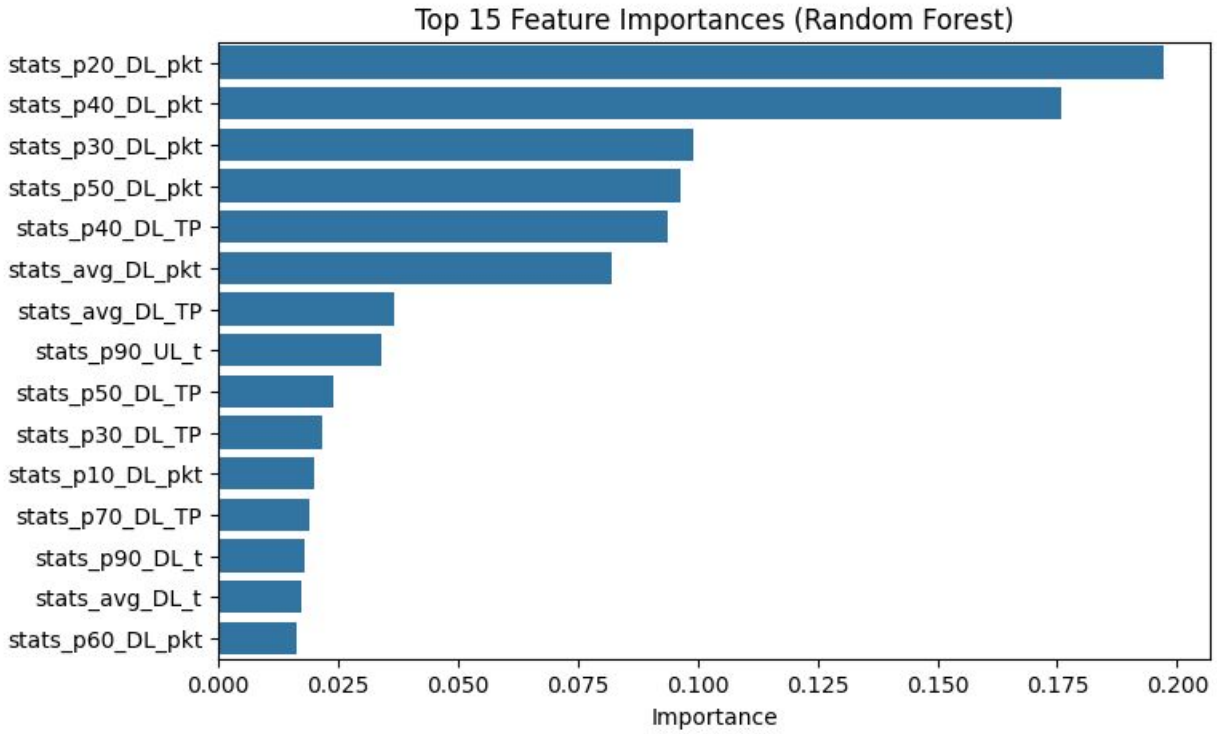
F1-score by feature visibility

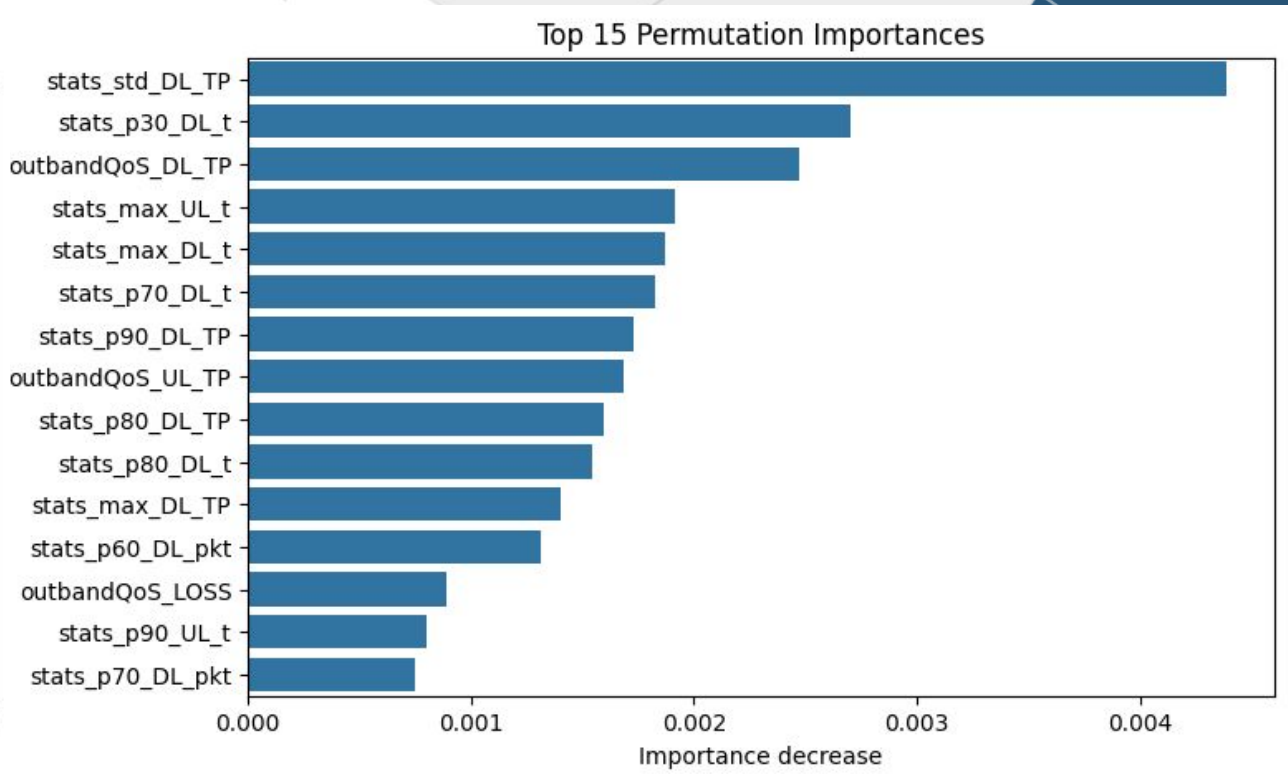
- **Minimal:** ~0.78
- **Operator:** ~0.89
- **Full:** ~0.90

Key Takeaway: Operator-level features achieve near-optimal performance with minimal added complexity.



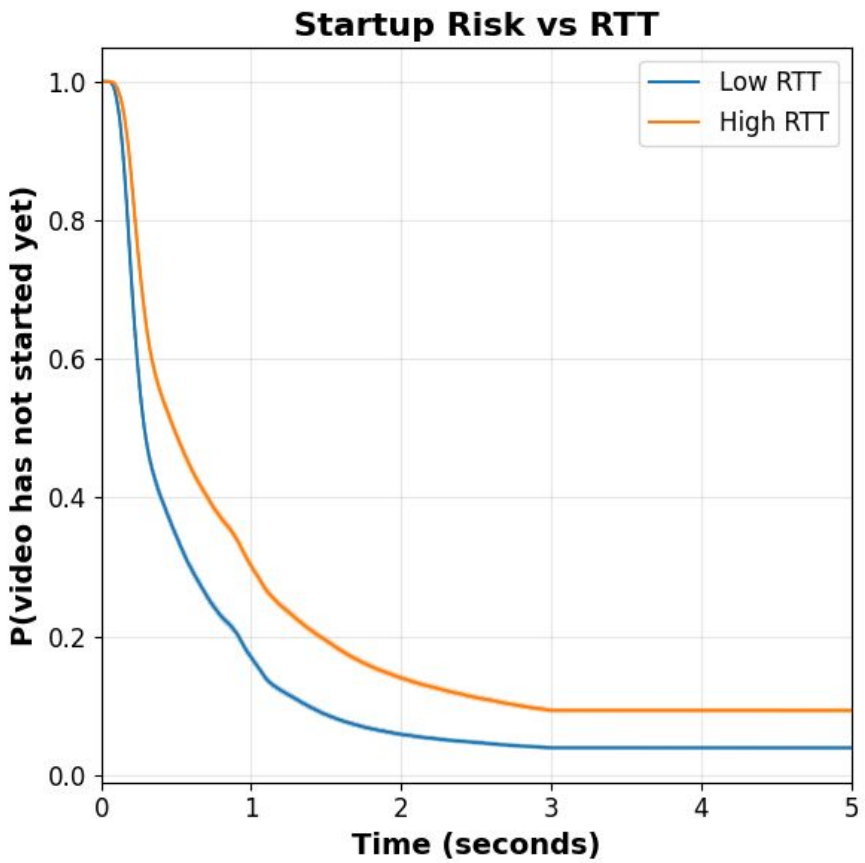
Feature Importance Analysis

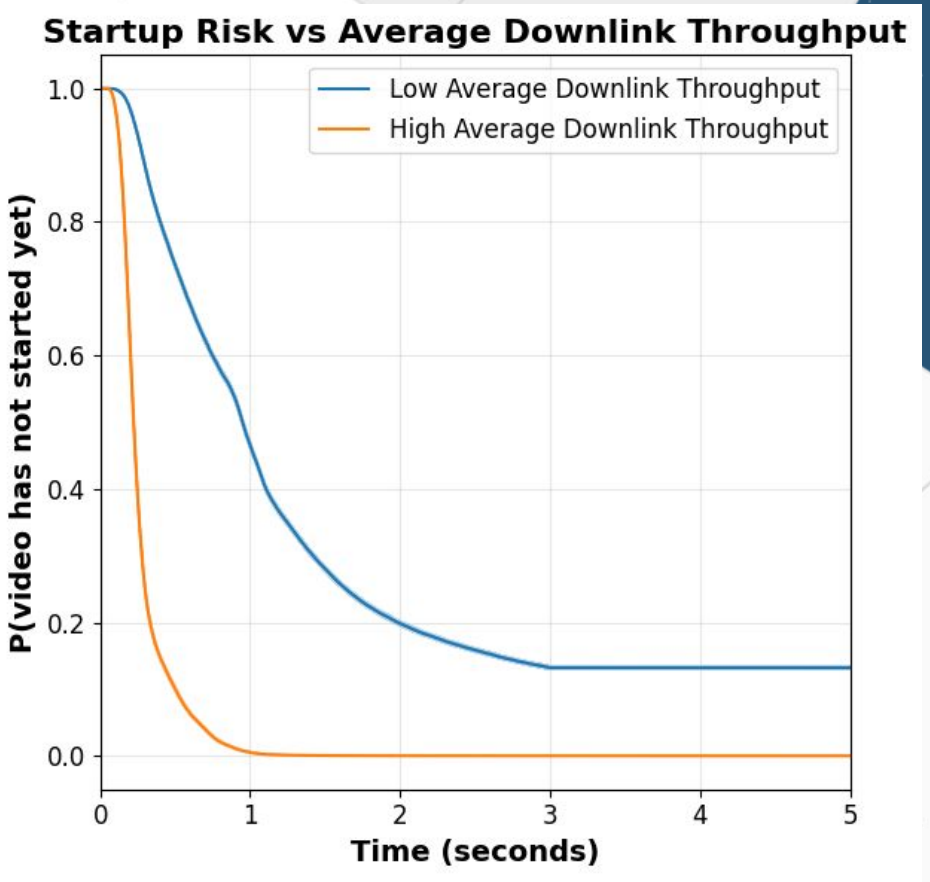




Key Takeaway: Startup delay is a bandwidth-limited phenomenon: poor downlink conditions dominate user-perceived startup time.

Survival Analysis - Network Conditions & Startup Risk *Average Downlink Throughput / RTT*





Key Takeaway : Startup success is dominated by bandwidth first, latency second : poor network conditions keep videos at risk far longer.

Startup Delay Prediction

Startup Delay Prediction - Problem Definition & Data Quality

The Prediction Task

Objective:

- Predict **continuous startup delay** (in seconds) for video sessions

Why This Matters:

- Complements classification (will it timeout?) with **quantitative prediction**
- Enables **precise resource allocation** (know exactly how much delay to expect)
- Users abandon after **>2 seconds** — every second counts (Krishnan & Sitaraman, 2012)

Regression Model & Experimental Design

Models

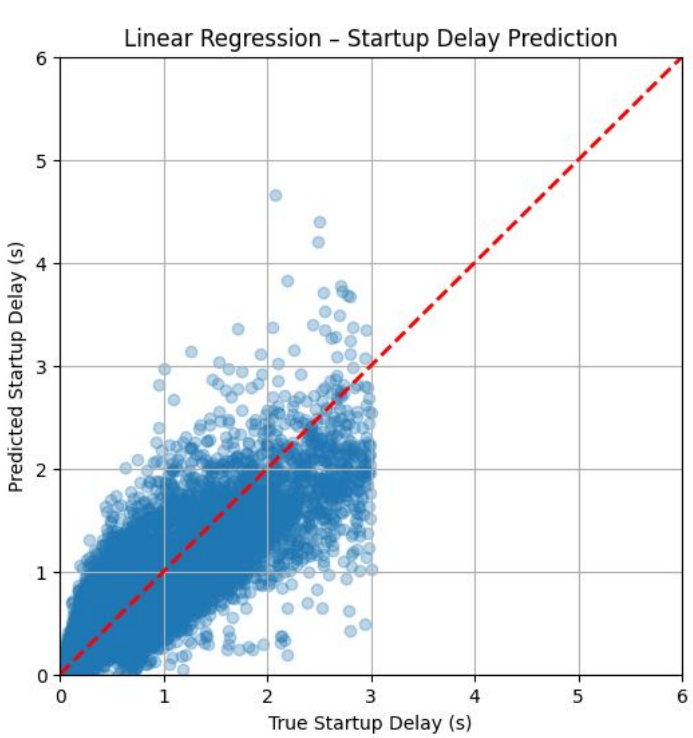
- Linear Regression
- Log-Linear Regression
- Random Forest

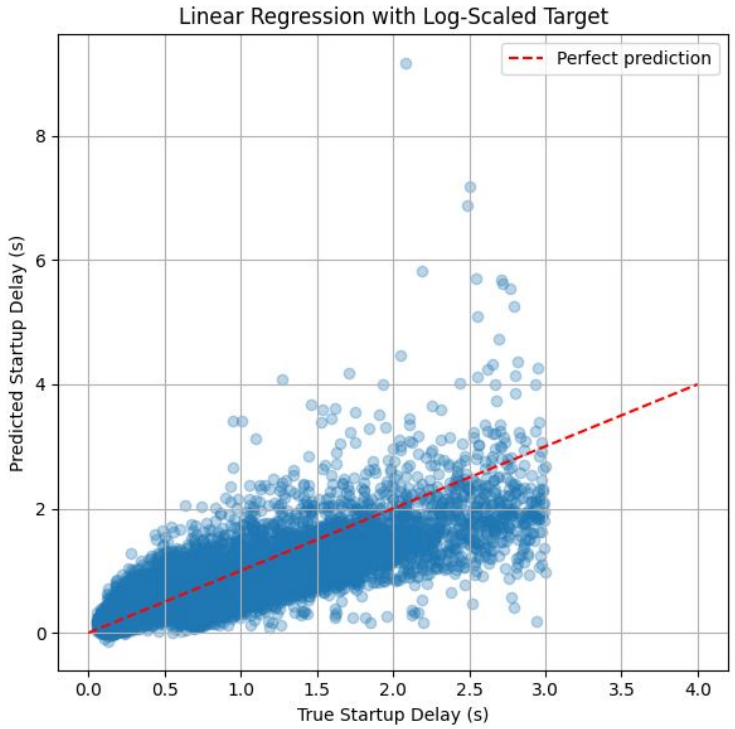
Validation

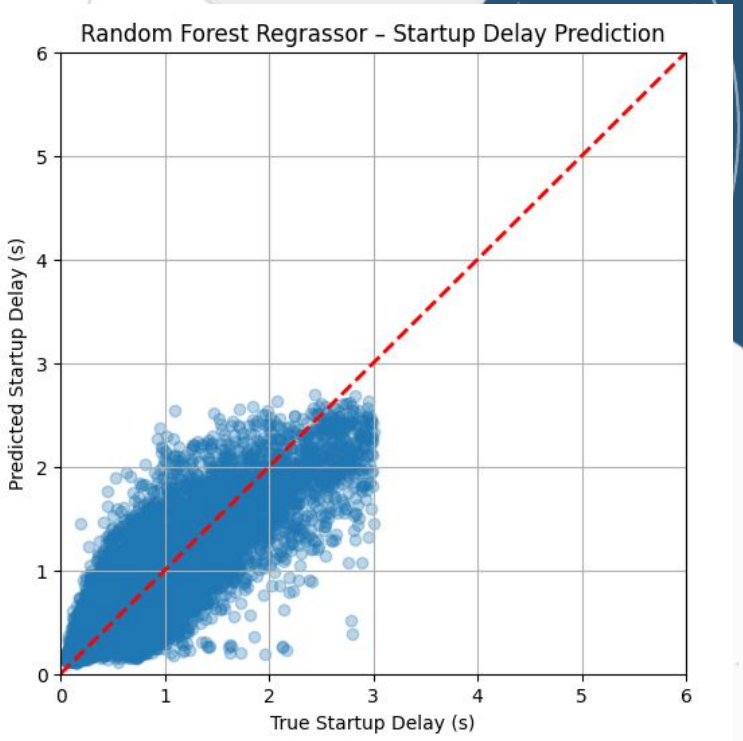
- 5-fold cross-validation
- Metric: RMSE (seconds)

Startup Delay Prediction

Model Performance Comparison: Linear vs. Tree-Based

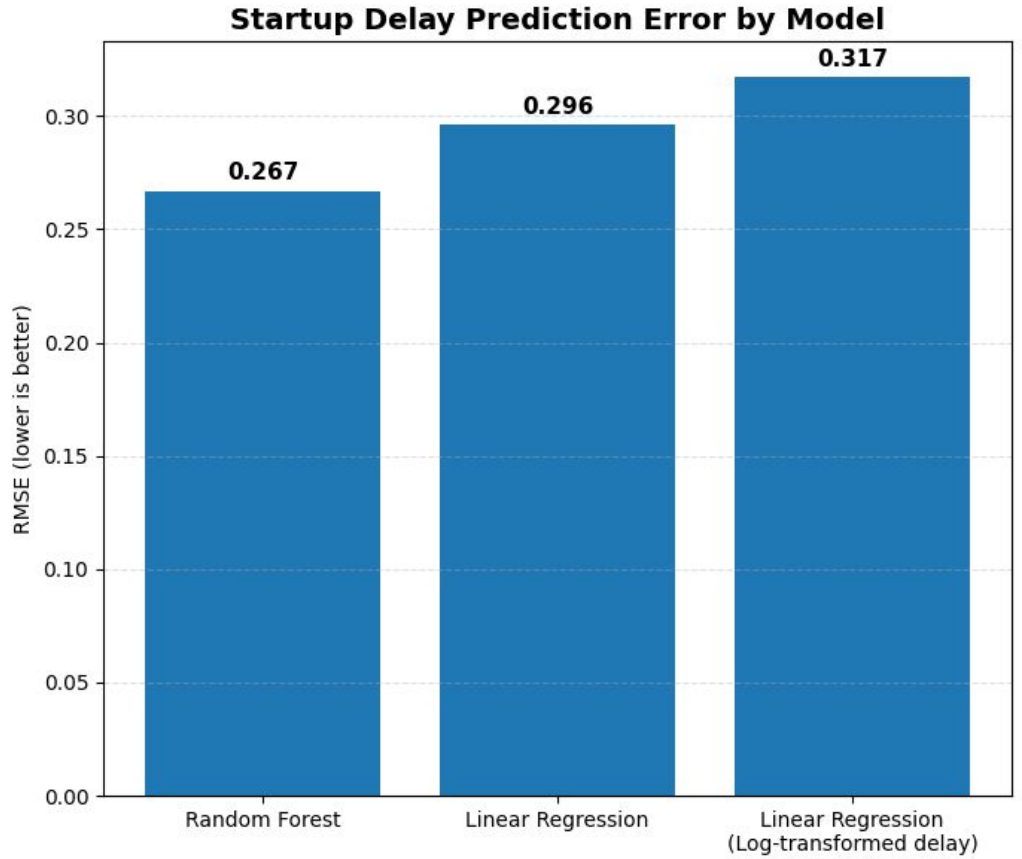




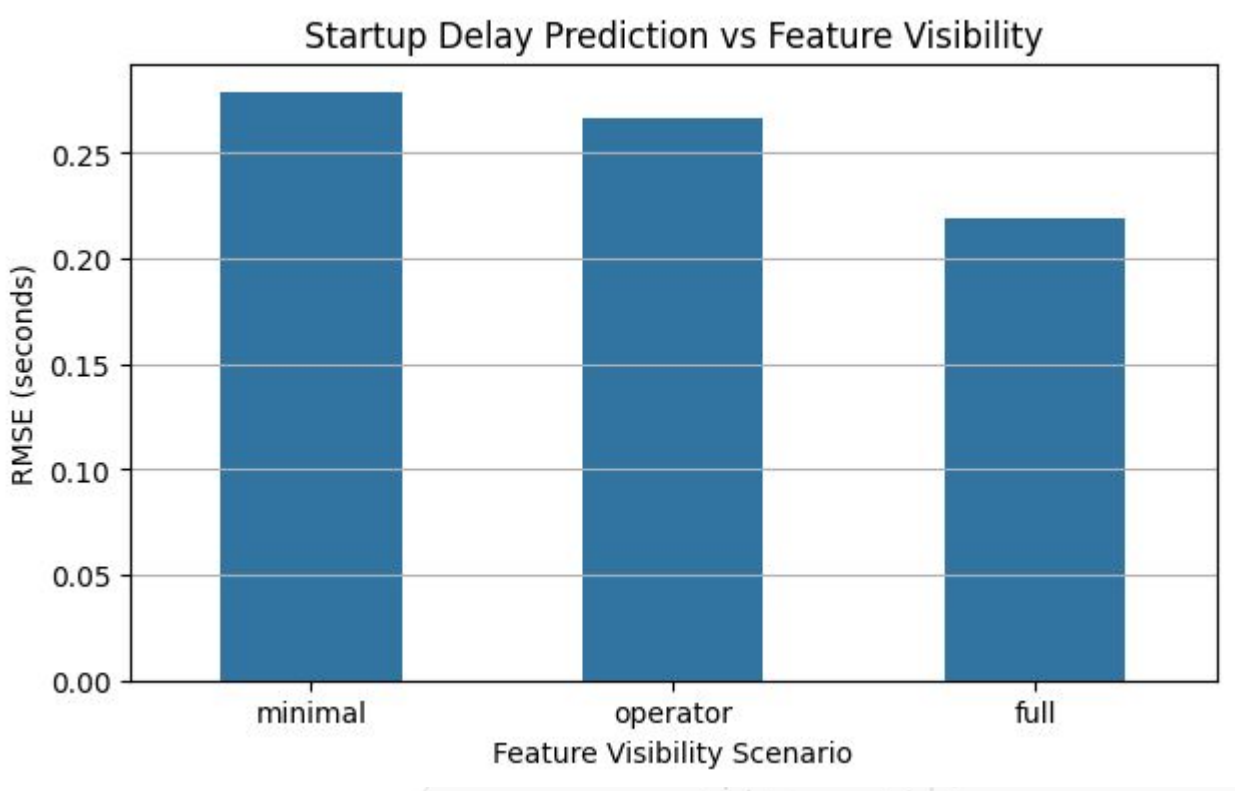


Key Takeaway : Tree-based models best capture startup delay, while linear models struggle with long-delay behavior.

Model Performance & Feature Set Comparison



Random Forest wins: 0.267s RMSE vs. 0.296s (linear) and 0.317s (log-linear). Non-linearity matters.



Diminishing returns: Full features only 18% better than minimal. Basic QoS surprisingly strong.

Random Forest Model Interpretability

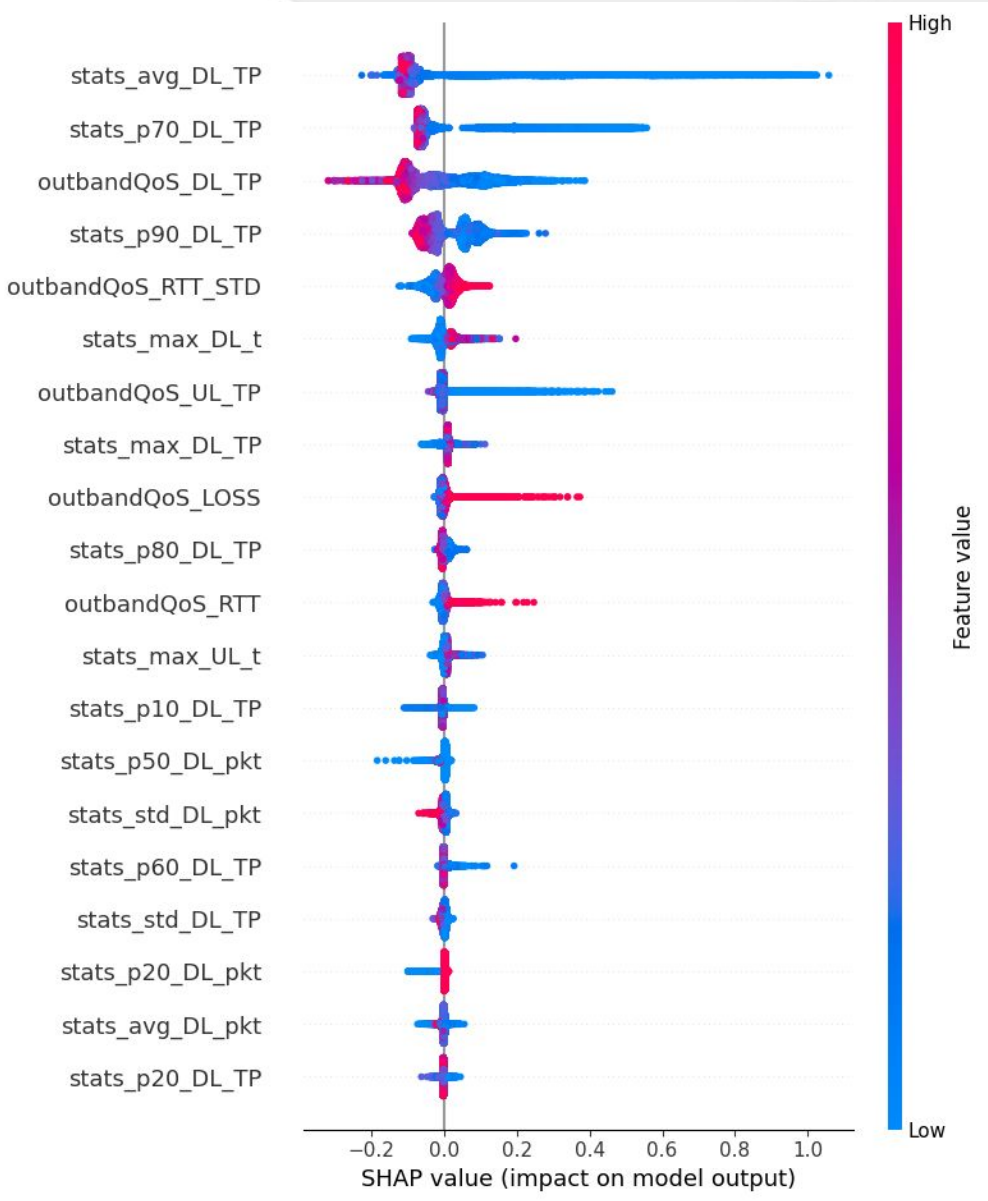
Coefficients vs. SHAP

Key Insights

- Startup delay is driven by bandwidth regimes
- Throughput tail behavior matter as much as averages
- Latency and loss act as amplifiers under degraded conditions
- Uplink effects are secondary but non-negligible

Key Takeaway

Startup delay emerges from non-linear interactions between bandwidth level, variability, and network degradation.



Linear Model Interpretability

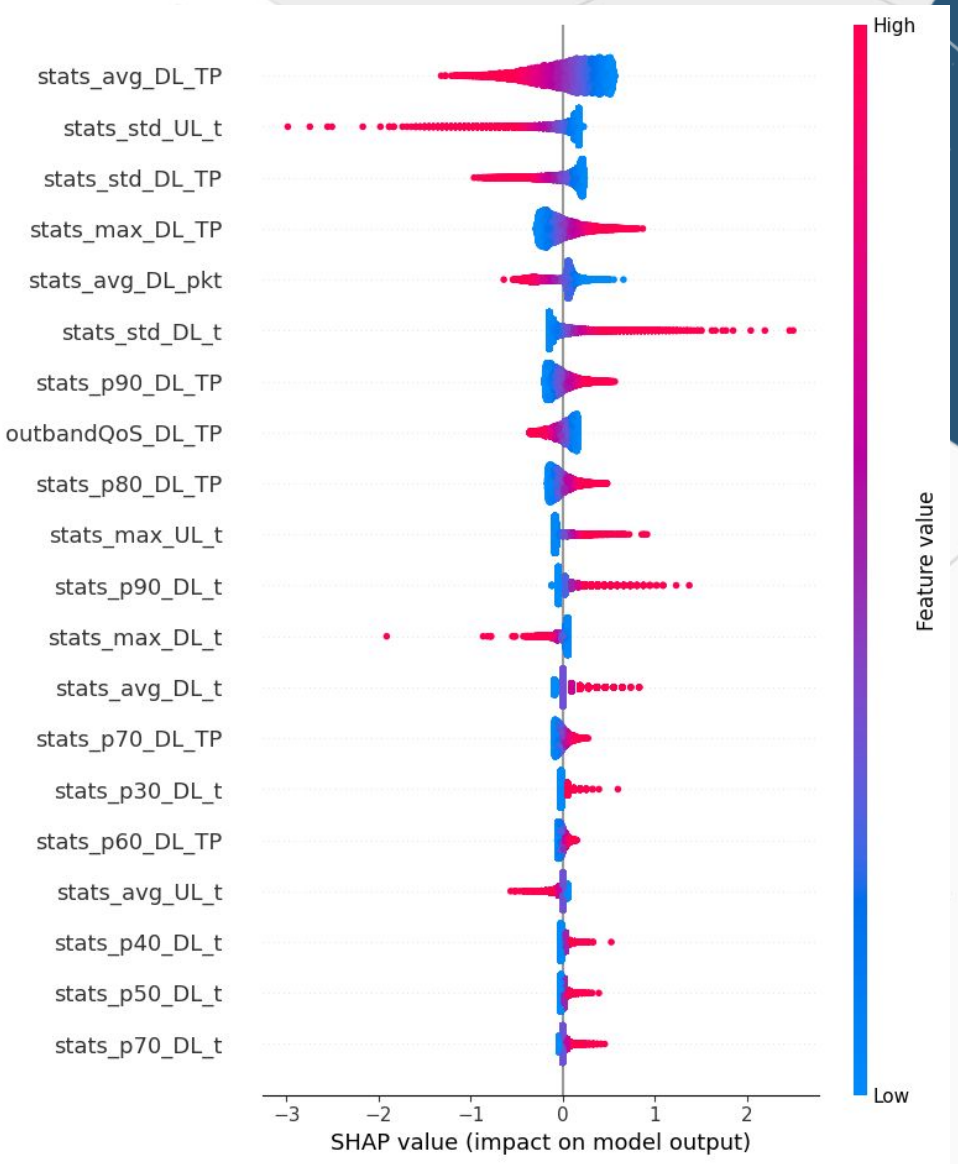
Coefficients vs. SHAP

Key Insights

- Average downlink throughput is the dominant driver of startup delay.
- Variability in throughput and delay matters, not just mean values.
- Extreme values disproportionately influence predictions.
- Effects are monotonic and additive, with no regime-dependent behavior.

Key Takeaway

The linear model explains startup delay mainly through average bandwidth and variability, but is dominated by extreme values and lacks interaction awareness.



QoE Score Prediction

Neural Network Architecture

- Scaled dataset
- Neural Network with 3 hidden layer
- Activation function: ReLU
- Feature sets used:
  - F_inband with high correlated features removed
  - F_outband: all features
- Adamm, learning rate = 0.001, batch_size= 32
- MSE for loss function

- Linear regression with the same dataset:
  - MSE: 0.5702
  - RMSE: 0.7551
  - MAE: 0.5175

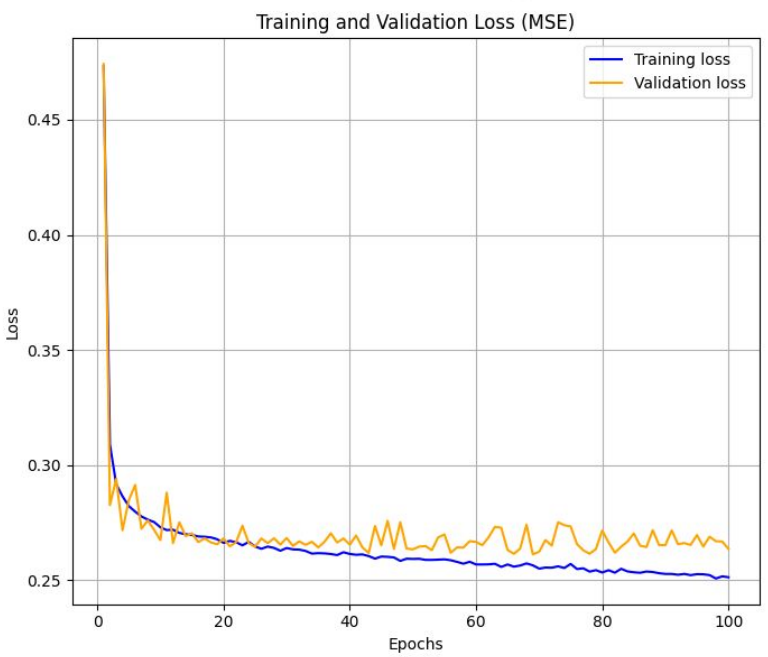
Layer (type)	Output Shape	Param #
expansion_layer (Dense)	(None, 128)	5,760
dropout_1 (Dropout)	(None, 128)	0
compression_layer (Dense)	(None, 64)	8,256
dropout_2 (Dropout)	(None, 64)	0
bottleneck_layer (Dense)	(None, 32)	2,080
output_layer (Dense)	(None, 1)	33

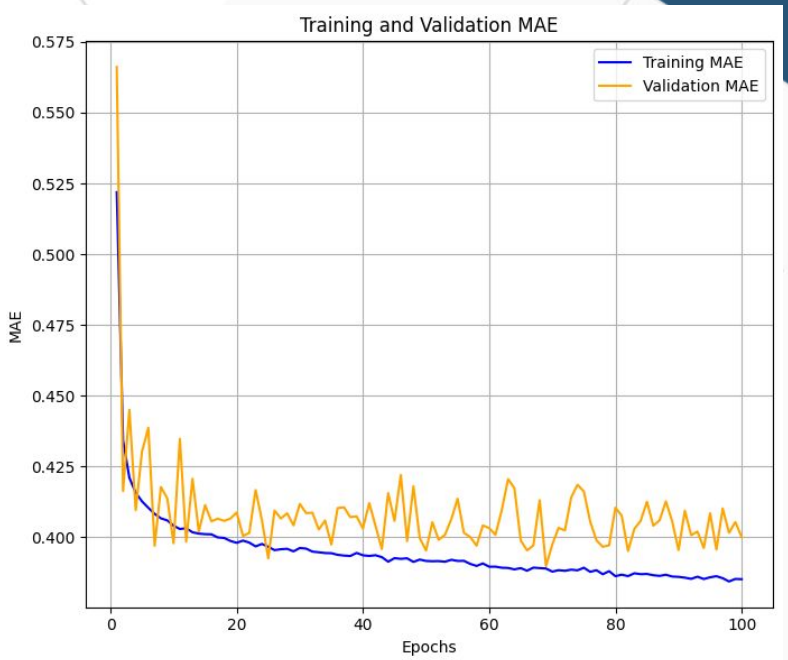
Total params: 16,129 (63.00 KB)
Trainable params: 16,129 (63.00 KB)
Non-trainable params: 0 (0.00 B)

Model Performance

- Test Set Evaluation
  - Test MSE: 0.2732
  - Test MAE: 0.4075
  - Test RMSE: 0.5227

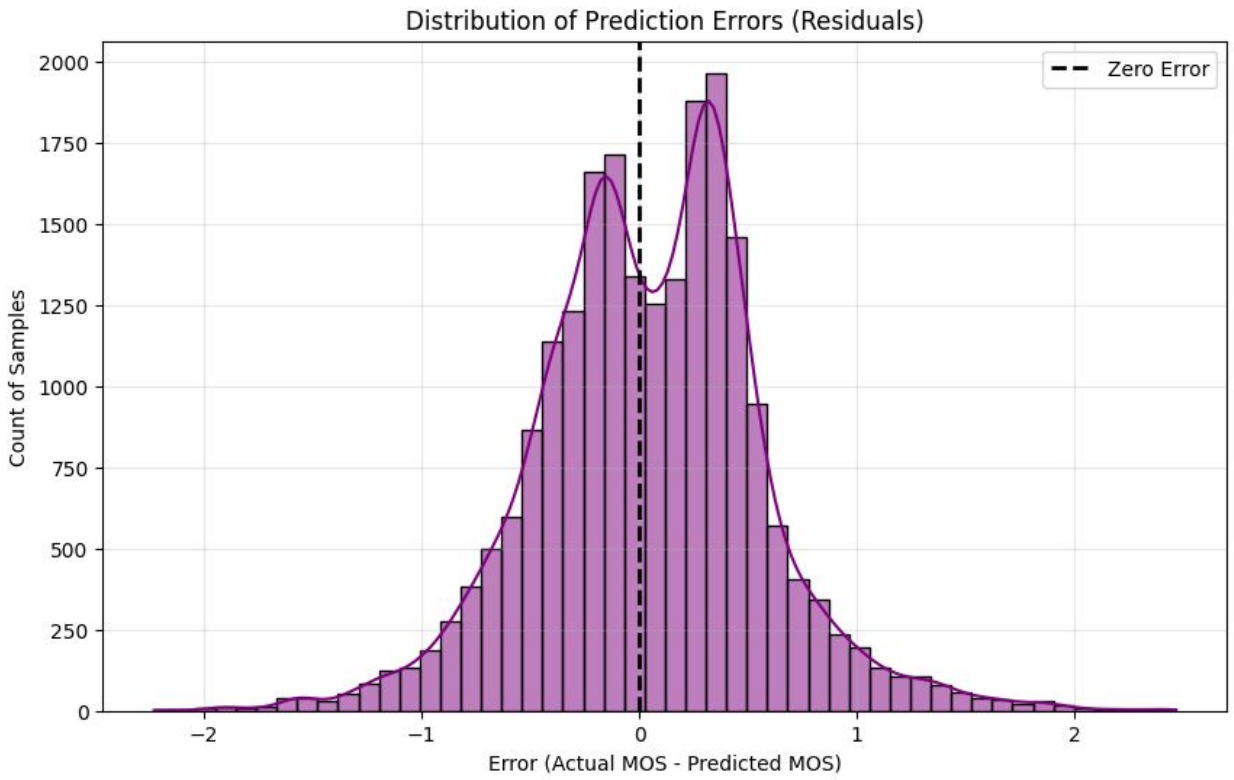
- K-fold (5 fold):
  - Average MSE: 0.2625 (+/- 0.0048)
  - Average MAE: 0.3970 (+/- 0.0044)
  - Average RMSE: 0.5124 (+/- 0.0047)

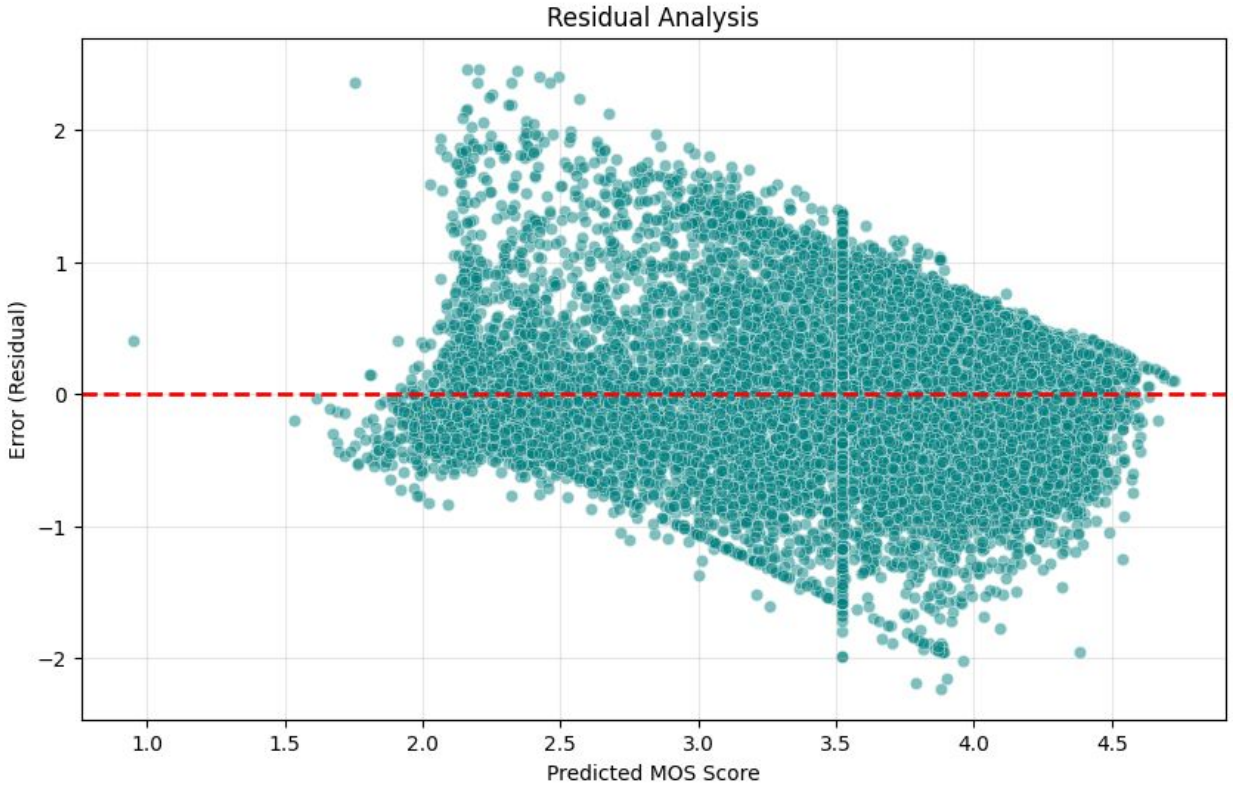




Model Performance

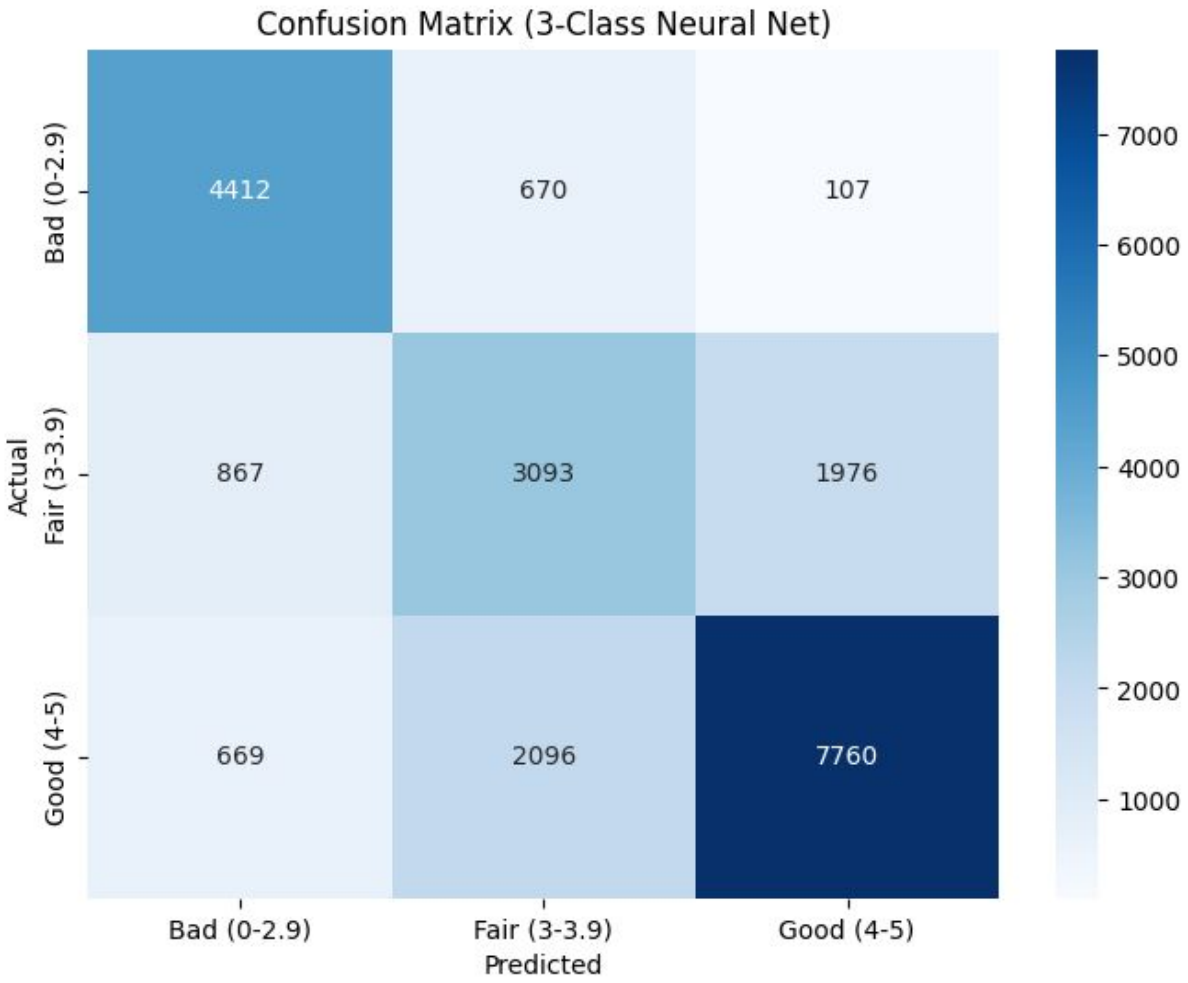
The model tends to underestimate actual low MOS and overestimate actual real MOS



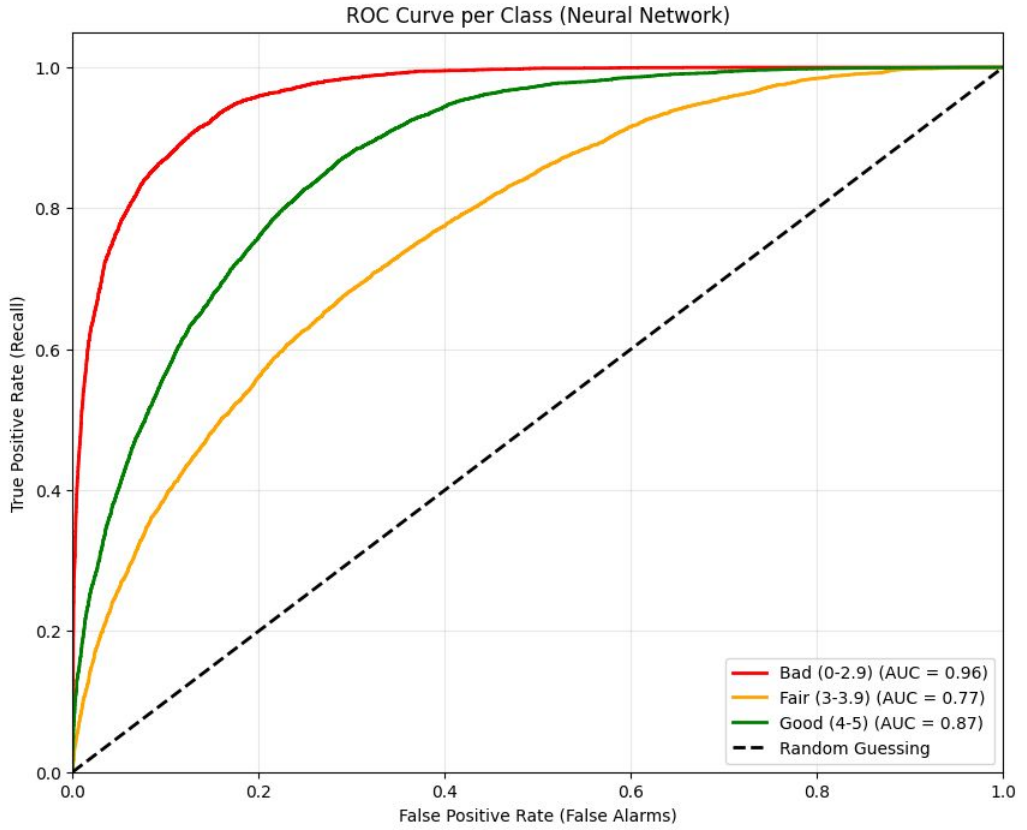


MOS Classification: Neural Network

The model still confuses between Fair and Good

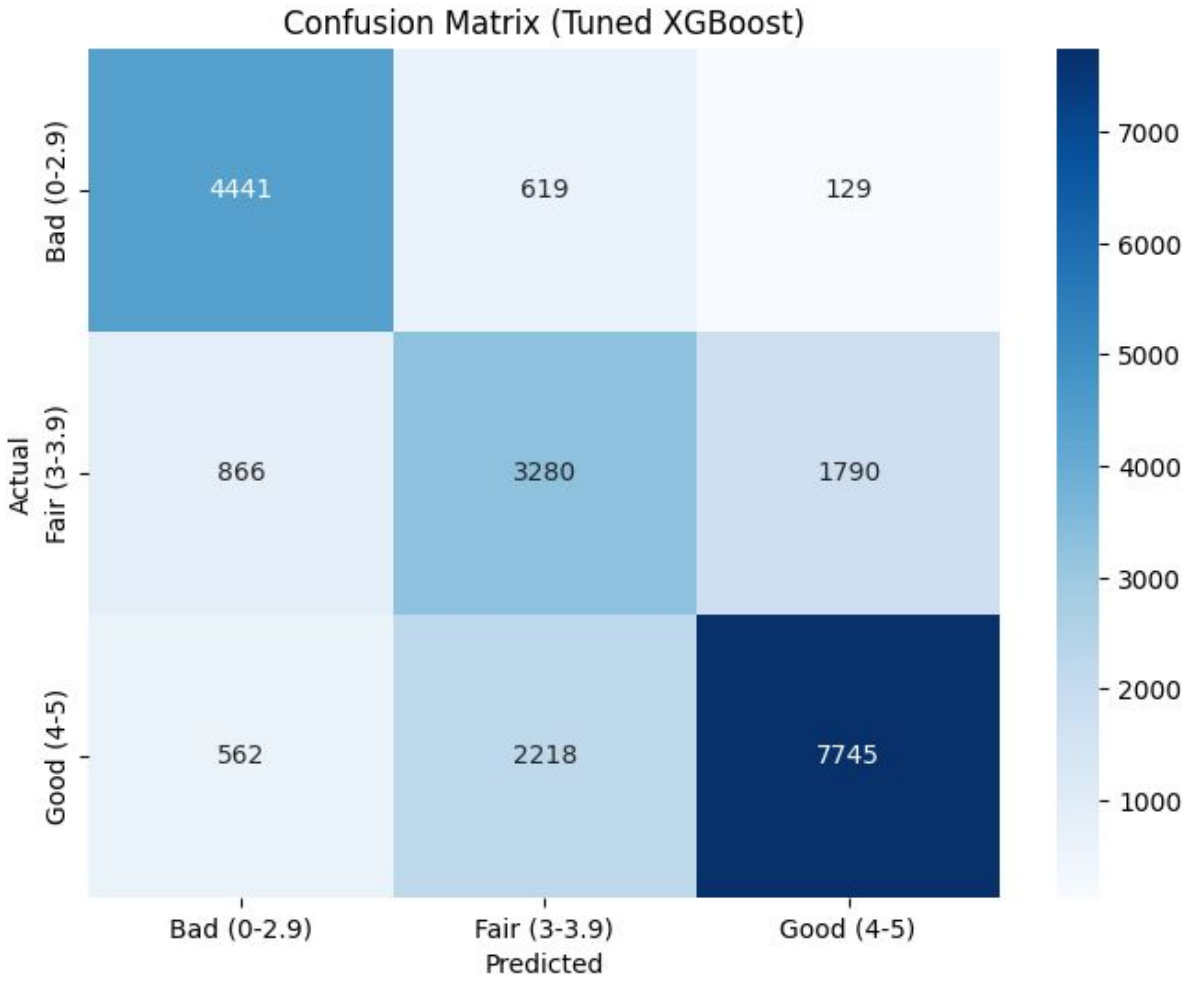



```
Classification Report:
              precision    recall  f1-score   support

 Bad (0-2.9)       0.74      0.85      0.79      5189
Fair (3-3.9)       0.53      0.52      0.52      5936
 Good (4-5)        0.79      0.74      0.76     10525
```

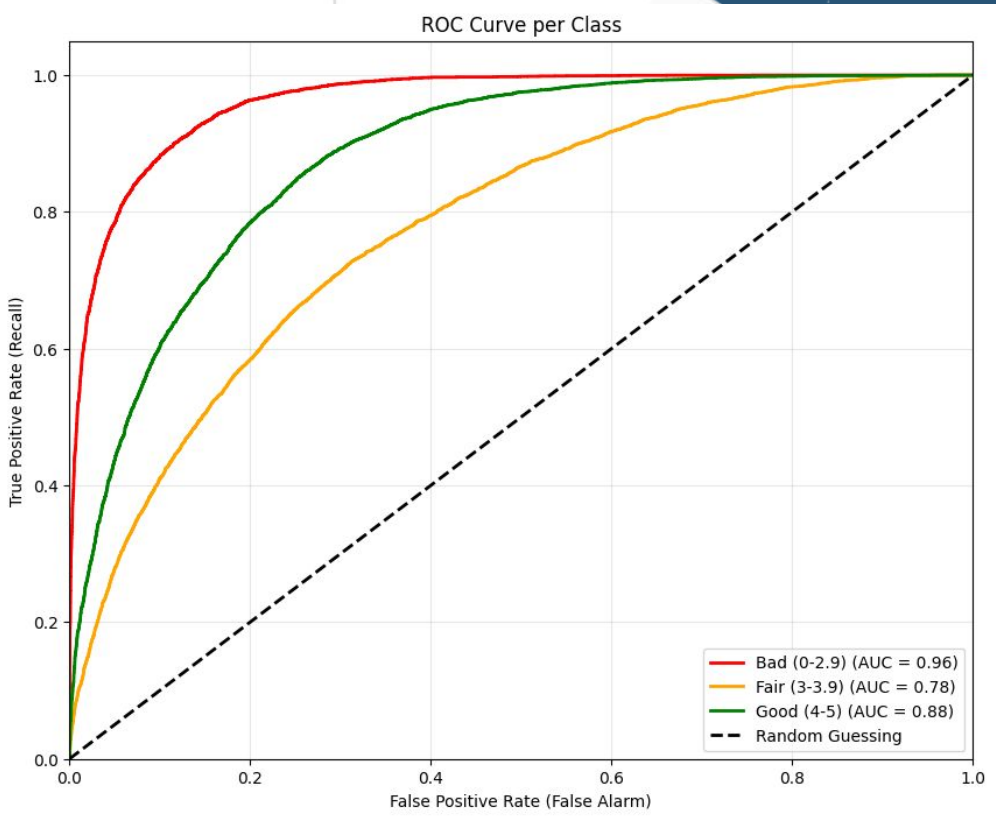


MOS Classification: XGBoost

Slightly better than Neural Network



Classification Report:	precision	recall	f1-score	support
Bad (0-2.9)	0.76	0.86	0.80	5189
Fair (3-3.9)	0.54	0.55	0.54	5936
Good (4-5)	0.80	0.74	0.77	10525



TELECOM SudParis

INSTITUT POLYTECHNIQUE DE PARIS

THANK YOU !

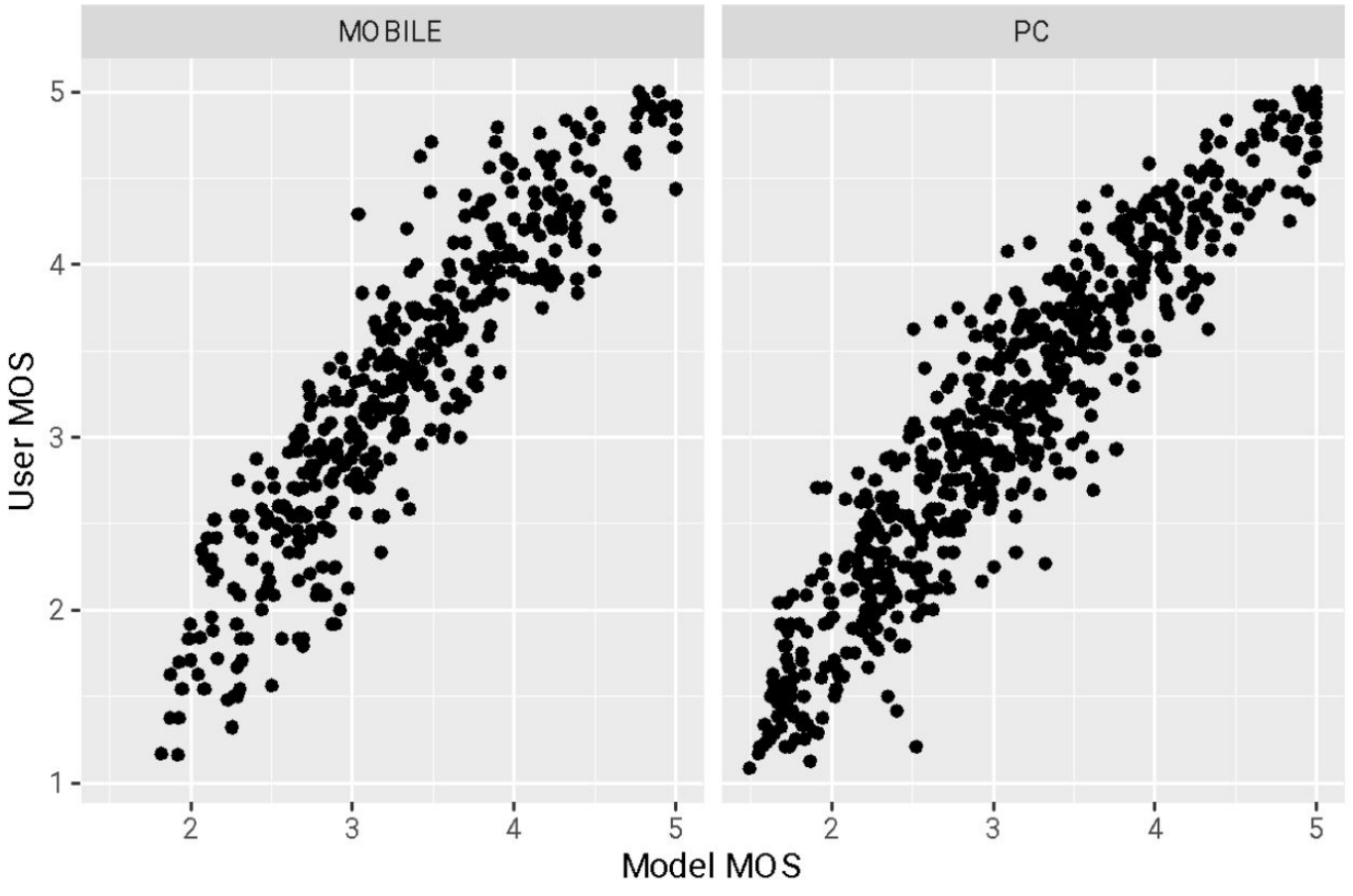
Institut Mines-Télécom

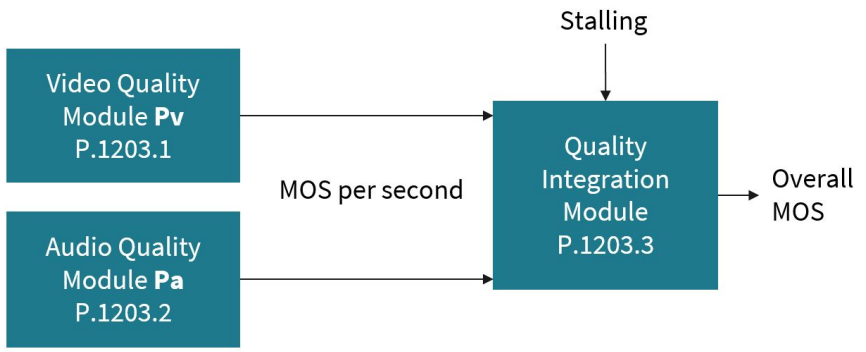
ITU-T P.1203

- Standardized algorithmic model used to measure QoE

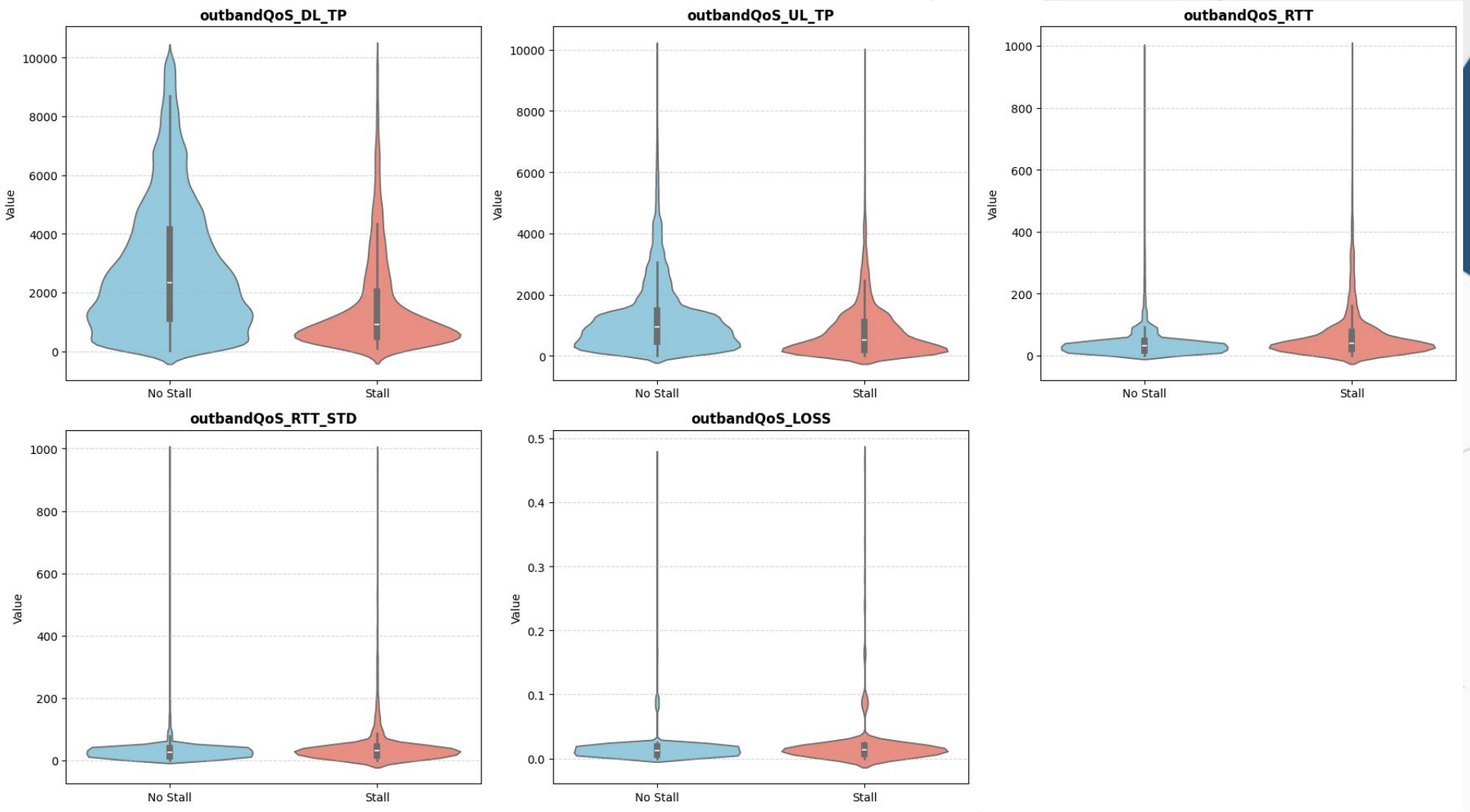
- P.1203.1 for Video and P.1203.2 for Audio
- P.1203.3: combine the scores and other factor (stalls, quality switch) for overall MOS

- Video stream data is ran through the algorithm.
  - Reads metadatas, frame header, bitstream, etc.





Data Exploration



Model Performance

With MAE as loss function, the problem still occurs

- Test MSE: 0.4168
- Test MAE: 0.4168
- Test RMSE: 0.5875