Machine Learning & Pattern Recognition (AI3011)

PLAYER AFFECTIVE STATE - DRIVEN GAMING EXPERIENCE MODULATION

Garvit Jain, Siddhartha Kumar, Vageesh Chandra Srivastava

THE PROBLEM

Most video games rely on static difficulty systems (Easy, Medium, Hard) or simple performance-based adjustments (score, time, health).

However, these approaches fail to account for the player's internal emotional state, which plays a critical role in engagement.

As a result:

- Players may feel bored when the game is too easy
- Players may feel frustrated or stressed when the game is too difficult
- Current boss/AI systems do not adapt to real-time affective states

How can we design a boss AI system that dynamically adjusts difficulty based on the player's real-time affective state inferred from gameplay behavior?

APPLICATIONS & IMPACT

Applications

- Commercial Video Games
  - Affective State-adaptive difficulty adjustment
  - Personalized gameplay experiences
  - Improved player retention
- Esports and Training Systems
  - Adaptive training bots
  - Affective states-aware practice environments

Impact

- Reduces player drop-off caused by frustration or boredom
- Enhances immersion through dynamic gameplay adaptation
- Enables highly personalized player experiences

Key Insight:
Instead of the player adapting to the game, the game adapts to the player.

LITERATURE REVIEW

1. Automated stress detection using keystroke and linguistic features: An exploratory study

Objective:
Determine whether stress can be detected using typing behavior alone

Methodology:
Participants performed typing tasks in two conditions:

- Baseline condition (relaxed typing)
- Stress condition (mental arithmetic + multitasking)

Researchers analyzed keystroke dynamics

Features Extracted:

- Key Hold Time (Dwell Time)
- Duration between key press and release
- Flight Time
- Time between consecutive key presses
- Typing Speed
- Backspace Frequency

Machine Learning Models:

- Decision Trees
- Support Vector Machines (SVM)

The system achieved approximately **75% accuracy** in detecting cognitive stress

Limitation:
Limited to typing tasks; does not generalize to complex interactive environments like games.

Vizer, L. M., Zhou, L., & Sears, A. (2009). Automated stress detection using keystroke and linguistic features: An exploratory study. International Journal of Human Computer Studies, 67(10), 870-886. https://doi.org/10.1016/j.ijhcs.2009.07.005

Important finding: Variability in typing rhythm is a strong indicator of stress.
We extend this idea by using keyboard + mouse telemetry in gameplay, which captures richer behavioral patterns.

LITERATURE REVIEW

2. Dynamic Difficulty Adjustment in Computer Games Through Real-Time Anxiety-Based Affective Feedback

Objective:
Develop a game system that adjusts difficulty based on player anxiety levels rather than performance metrics

Experimental Setup:
Two versions of the game were tested:

- Traditional Version
  - Difficulty changes based on score.
- Emotion-Adaptive Version
  - Difficulty adjusts based on physiological signals.

Proposed Solution:
Adjust game difficulty using real-time physiological signals (ECG, GSR, EMG).

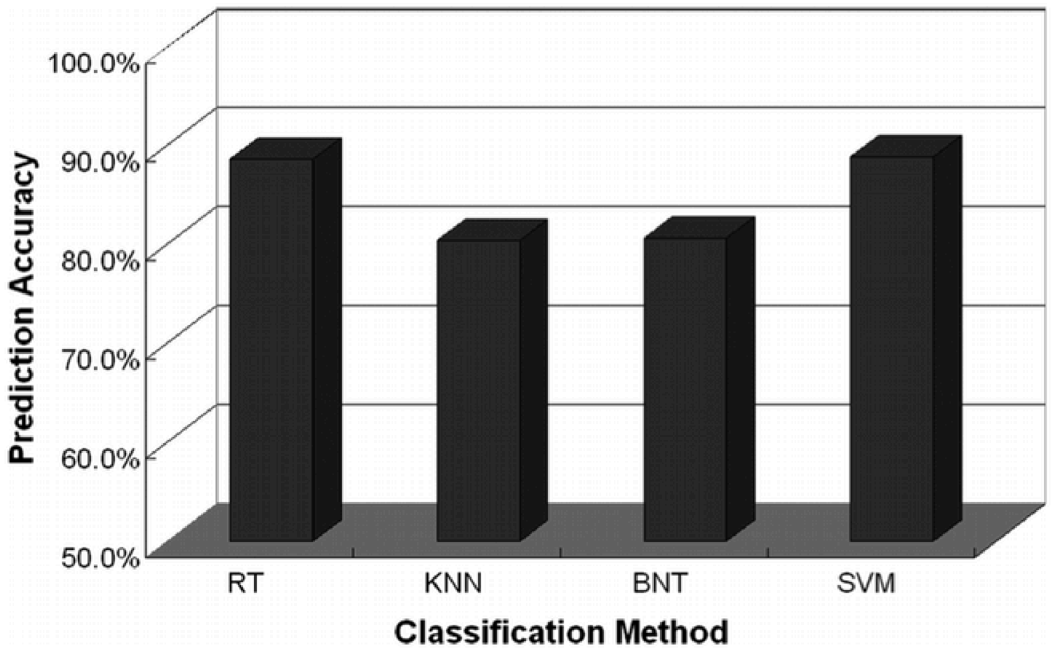
Sensors Used

- ECG (Electrocardiogram)
  - Measures heart rate
- GSR (Galvanic Skin Response)
  - Measures skin conductance
- EMG (Electromyography)
  - Measures muscle tension

Players using the affective system:

- Reported higher engagement (85% Accuracy)
- Remained in the flow state longer
- Played for longer durations

Limitation:
Requires expensive physiological sensors.
We adopt the affective feedback loop, but replace physiological sensors with behavioral telemetry, making the system more scalable and practical.



Liu, C., Agrawal, P., Sarkar, N., & Chen, S. (2009). Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. International Journal of Human-Computer Interaction.

LITERATURE REVIEW

3. Circumplex Model of Affect

Concept:

The Circumplex Model of Affect represents emotions along two dimensions:

- Valence (positive ↔ negative)
- Arousal (low ↔ high)

This provides a structured way to categorize emotional states.

Proposed Idea

- Emotions are not discrete but exist in a continuous space
- Different affective states can be grouped based on similar arousal and valence levels

Key Insight

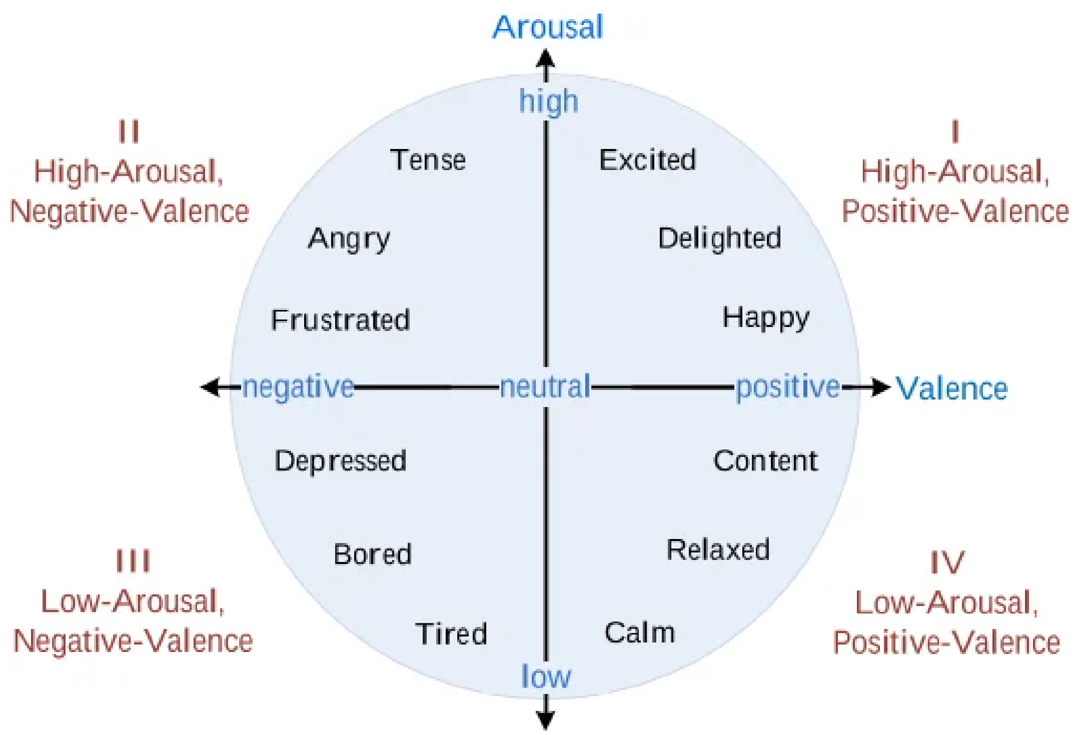
- Emotions with similar behavioral patterns often lie close in this space
- Helps explain overlap between emotional states

Limitations

- The model is theoretical and does not directly map to gameplay features
- Requires interpretation to connect with real-world signals

Relation to Our Work (Extension)

- We use this model to define and group player affective states
- Selection of states:
    - Flow / Enjoying → positive valence
    - Bored → low arousal
    - Frustration → high arousal, negative valence
- Helps explain why some states overlap in gameplay behavior



Russell, James. (1980). A circumplex model of affect. Journal of Personality and Social Psychology. 39. 1161-1178. 10.1037/h0077714.

LITERATURE REVIEW

4. Flow: The Psychology of Optimal Experience

Concept:

Flow Theory states that optimal experience occurs when:

Challenge ≈ Skill

Proposed Idea

- Player experience depends on balance between challenge and skill
- Different imbalance states lead to:
  - Boredom → challenge too low
  - Flow → balanced
  - Frustration/Anxiety → challenge too high

Key Insight

- Maintaining flow leads to:
  - higher engagement
  - better user experience

Limitations

- Does not provide a method to detect player state automatically
- Requires external systems to estimate player experience

Relation to Our Work (Extension)

- We use Flow Theory to:
  - interpret predicted affective states
  - map them to challenge-skill imbalance
- Forms the basis for adaptive difficulty logic

Csikszentmihalyi, Mihaly. (1990). Flow: The Psychology of Optimal Experience.

LITERATURE REVIEW

5. Dynamic Difficulty Adjustment (DDA)

Hunicke, Robin, et al. MDA: A Formal Approach to Game Design and Game Research. Northwestern University, 2004.

Concept:
Dynamic Difficulty Adjustment (DDA) enables games to continuously adapt challenge levels in real time based on player behavior.

How DDA Works (Key Mechanism)

- The system monitors player performance metrics such as:
  - score
  - health
  - success/failure rate
- Based on these signals, the game modifies difficulty parameters dynamically, such as:
  - enemy strength
  - spawn rate
  - game speed

This creates a feedback loop:
Player Performance → Difficulty Adjustment → Updated Gameplay

Key Contribution of the Paper

- Demonstrates that real-time adaptive systems improve engagement
- Shows that difficulty should be continuously adjusted, not fixed
- Introduces the idea of maintaining a balance between challenge and skill

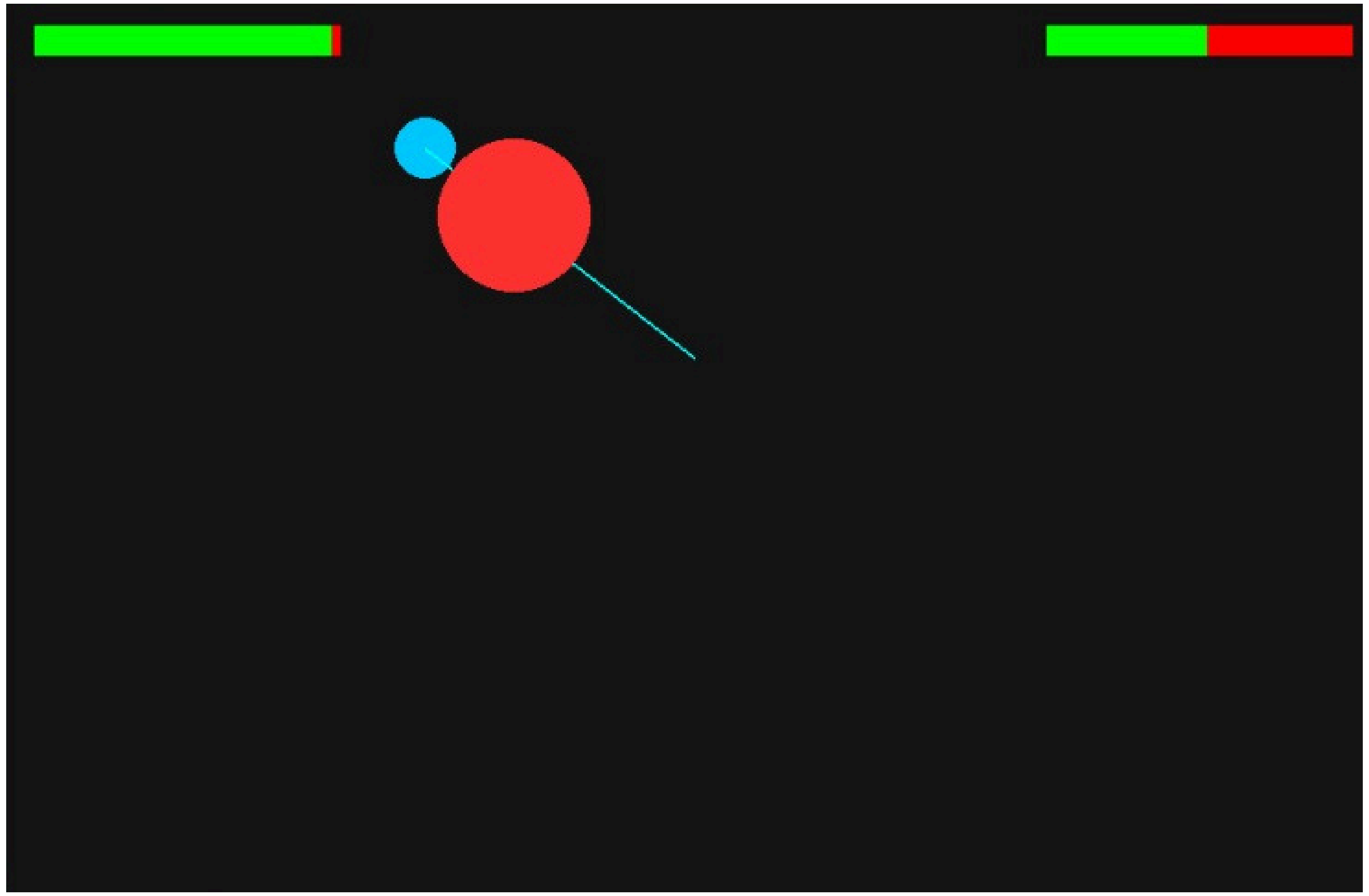
Limitations

- Relies only on performance metrics
- Does not consider player emotional or affective state
- May misinterpret performance (e.g., skilled but frustrated player)

Relation to Our Work (Extension)

- We adopt the same real-time adaptive feedback loop from DDA
- Instead of using performance metrics, we use:
- gameplay + keyboard/mouse telemetry
- machine learning to predict player affective state
- Our system:
  - Player Behavior → Affective State Prediction → Difficulty Adjustment

DATASET COLLECTION



Game Environment:
We developed a custom boss-fight game to collect gameplay telemetry.

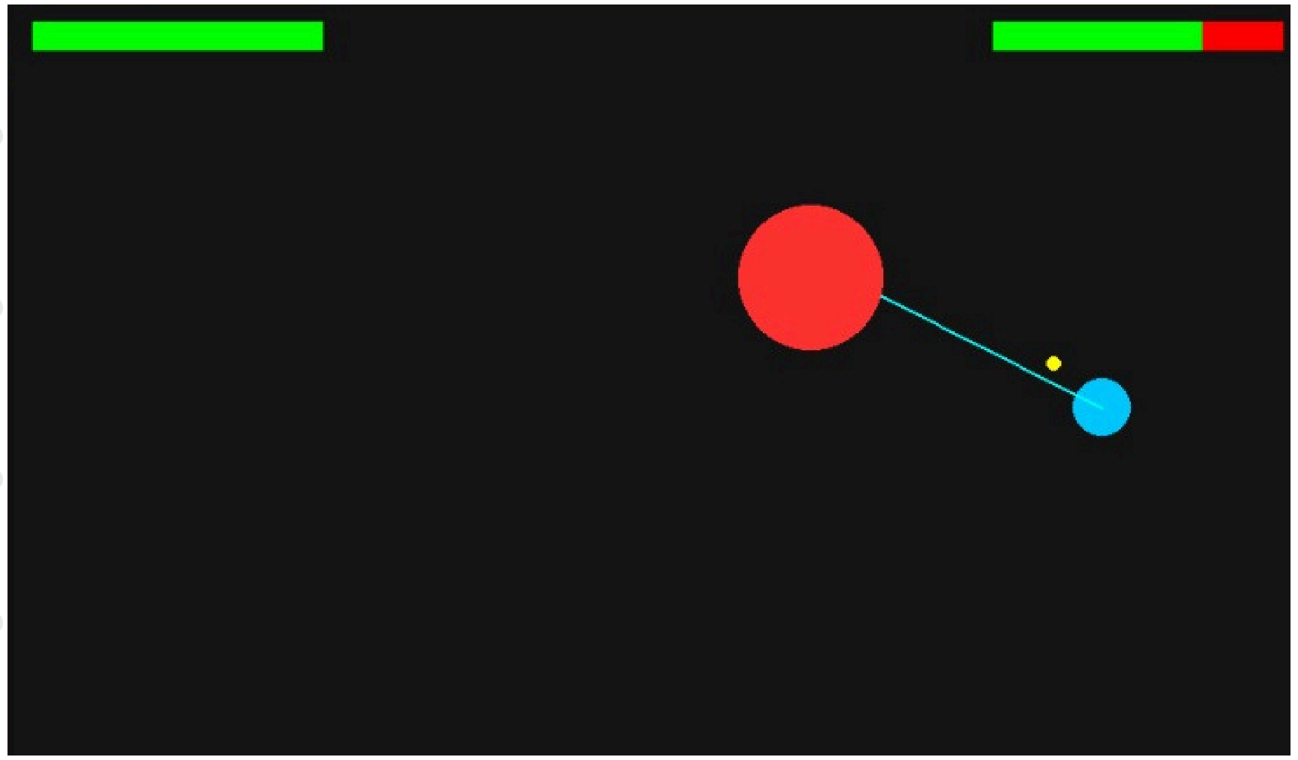
Game characteristics:

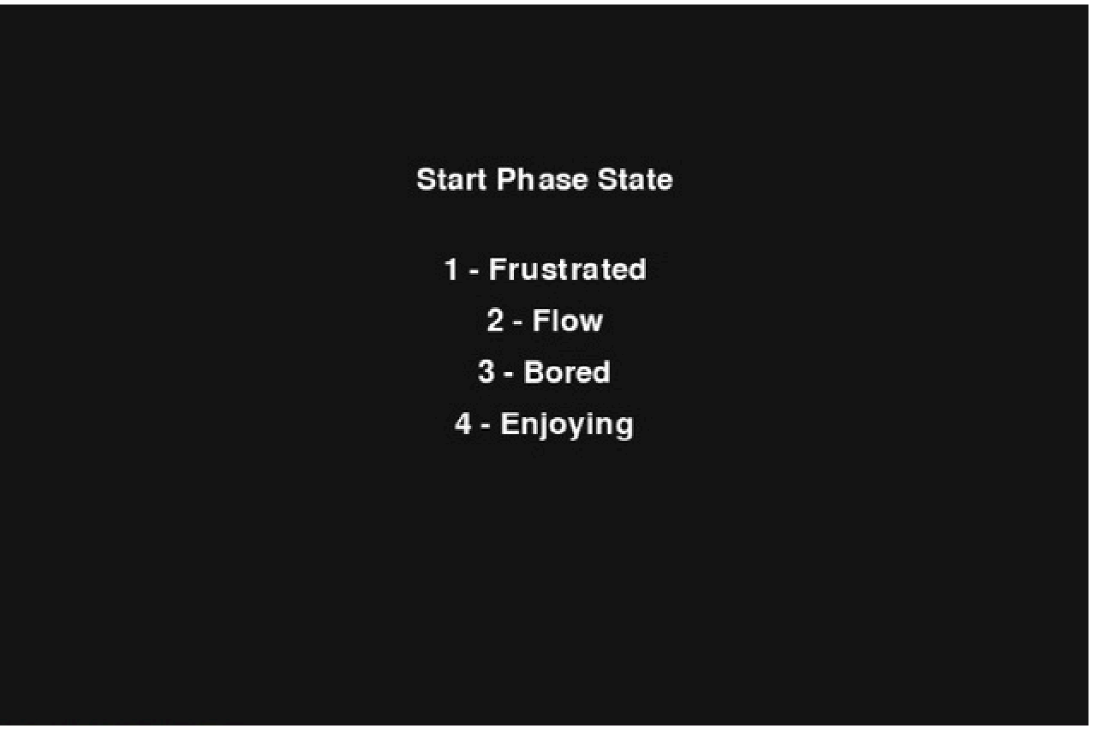
- Player vs Boss combat scenario
- Player controls movement using WASD keys and attacks using mouse clicks
- Multiple difficulty levels (Easy, Normal, Medium, Hard, Extreme)
- Player behavior and gameplay telemetry recorded in fixed 2-second intervals

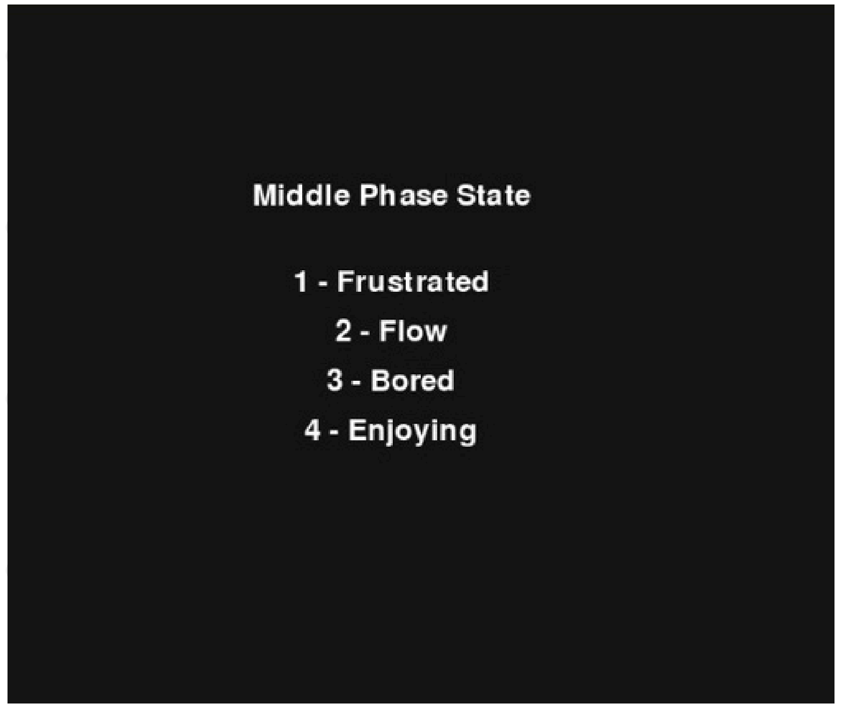
Ethical Considerations:

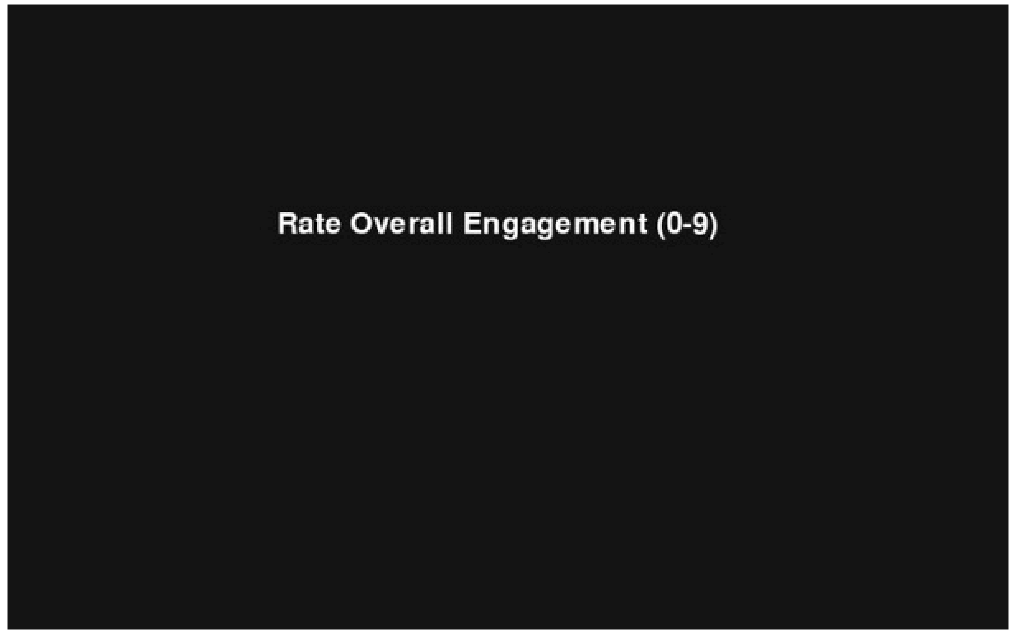
- Players were informed about data collection before gameplay
- Explicit consent was obtained prior to participation
    - Game only collects data when consent is given
- No personally identifiable information was collected

SNAPSHOTS OF THE GAME









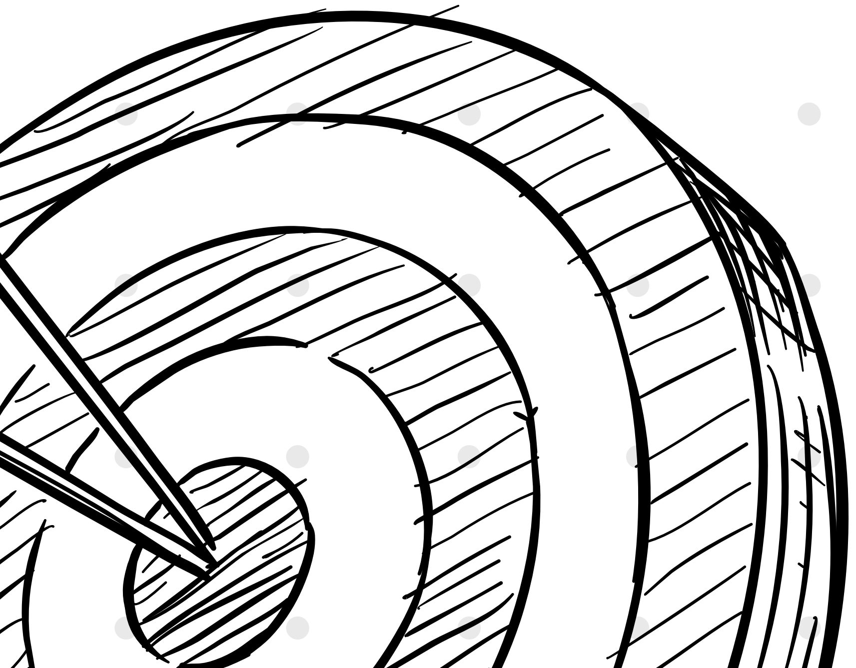
FEATURES EXTRACTED

Dataset Overview:

~2200 data samples collected

Collected from aroun 35 people

Total Games played per person: 5-6



Gameplay data is collected in fixed 2-second windows during each match. Each window generates one dataset entry.

Raw Features:

- shots_fired
- shots_hit
- movement_presses
- direction_changes
- distance_moved
- proximity_time
- damage_taken
- damage_dealt
- player_health
- low_health_time
- survival_time

Derived Features:

- accuracy = shots_hit / shots_fired
- click_freq = shots_fired / time_window
- avg_mouse_speed
- mouse_var
- aim_error
- idle_ratio = idle_time / window_duration
- aggression_index = click_freq × proximity_time

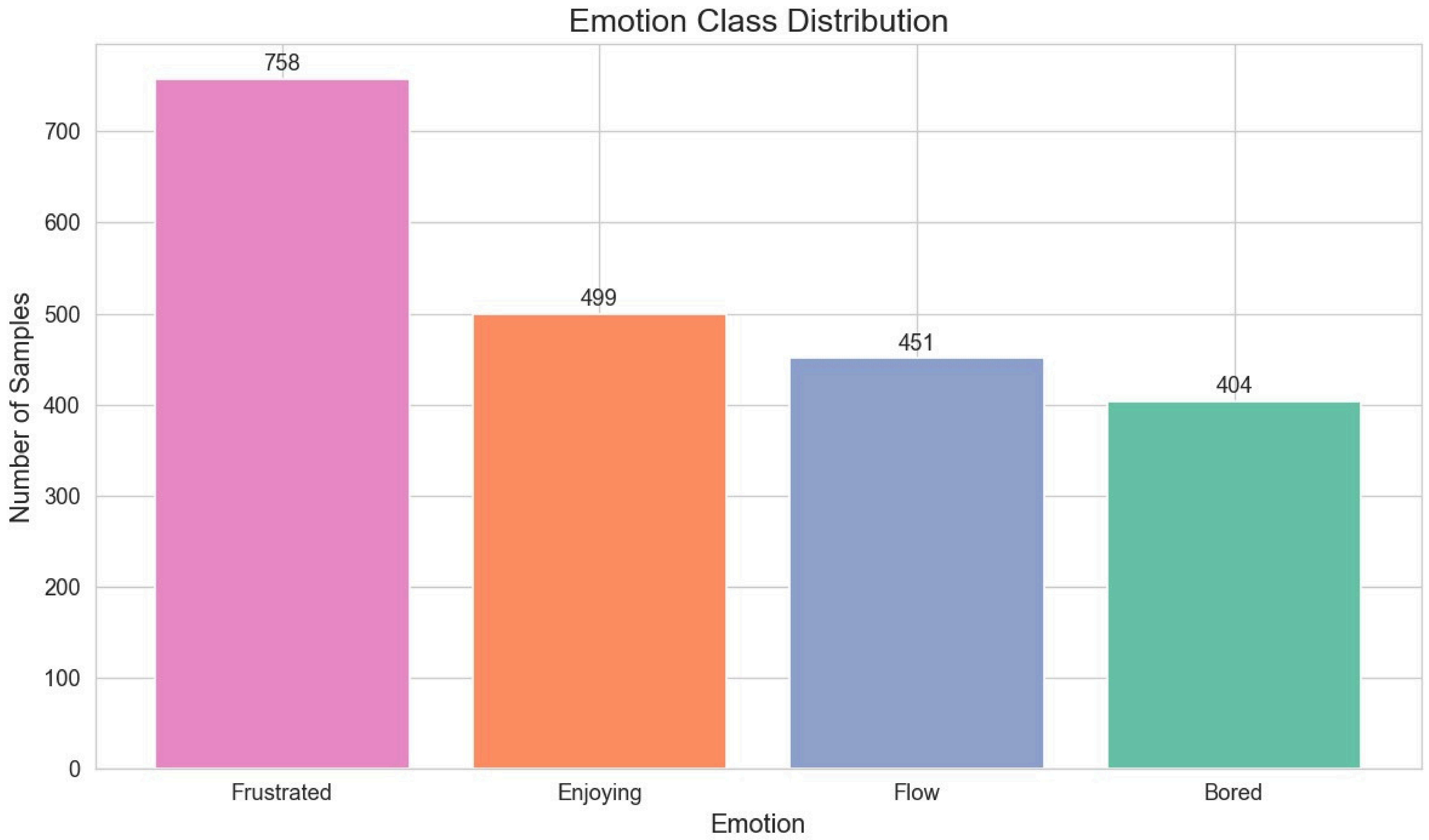
Metadata:

- match_id
- difficulty
- phase
- result
- engagement_score
- emotion

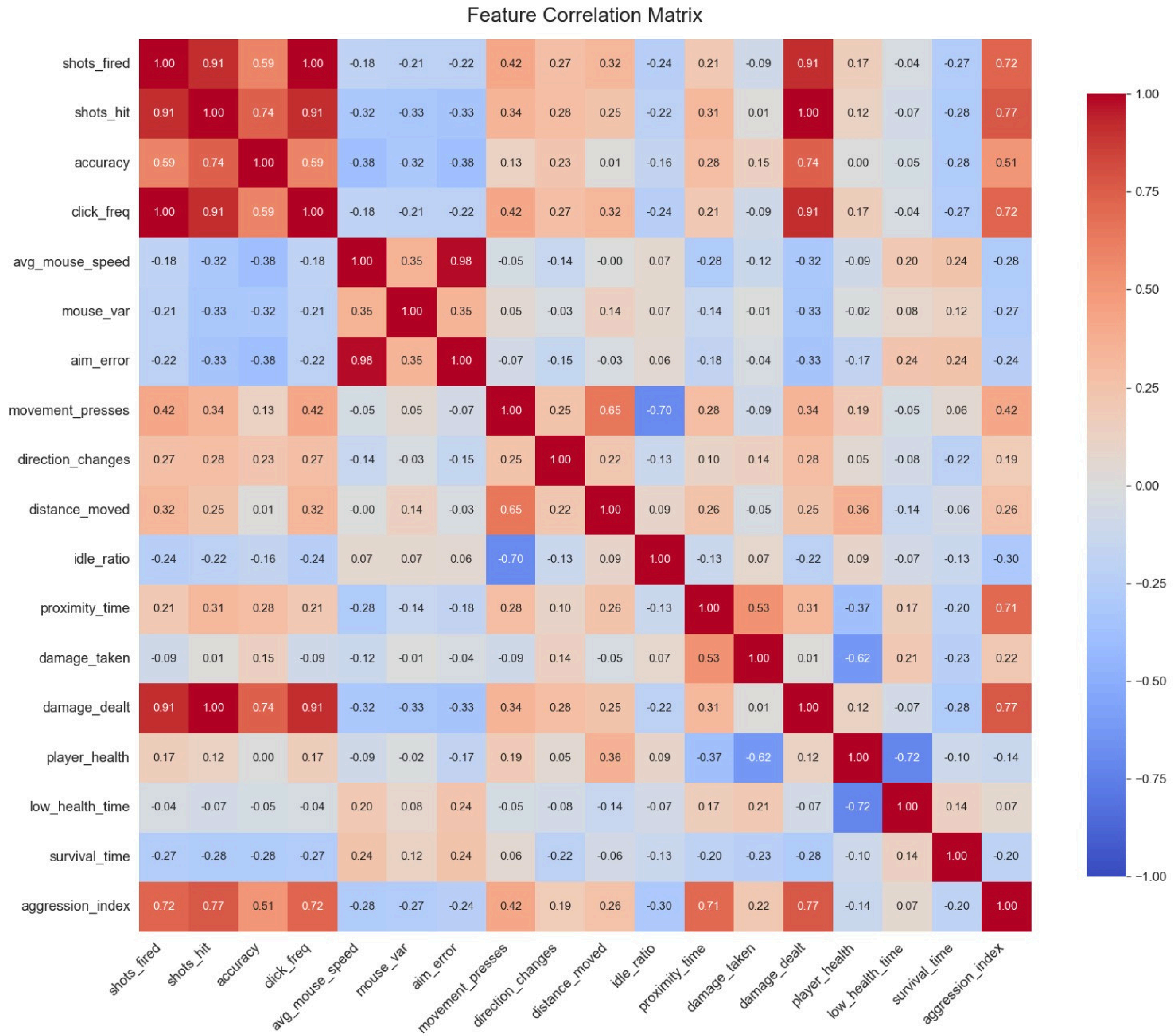
EMOTION LABEL COLLECTION

- After each gameplay session, players report their affective state
- Labels collected across 3 phases:
  - Beginning
  - Middle
  - End
- Final affective states used:
  - Flow
  - Bored
  - Enjoying
  - Frustration

Additional Feedback
Player provides an engagement score (1–10)



Emotion Class Distribution
758
499
451
404
Number of Samples
0
100
200
300
400
500
600
700
Frustrated
Enjoying
Flow
Bored
Emotion


Feature Correlation Matrix
shots_fired 1.00 0.91 0.59 1.00 -0.18 -0.21 -0.22 0.42 0.27 0.32 -0.24 0.21 -0.09 0.91 0.17 -0.04 -0.27 0.72
shots_hit 0.91 1.00 0.74 0.91 -0.32 -0.33 -0.33 0.34 0.28 0.25 -0.22 0.31 0.01 1.00 0.12 -0.07 -0.28 0.77
accuracy 0.59 0.74 1.00 0.59 -0.38 -0.32 -0.38 0.13 0.23 0.01 -0.16 0.28 0.15 0.74 0.00 -0.05 -0.28 0.51
click_freq 1.00 0.91 0.59 1.00 -0.18 -0.21 -0.22 0.42 0.27 0.32 -0.24 0.21 -0.09 0.91 0.17 -0.04 -0.27 0.72
avg_mouse_speed -0.18 -0.32 -0.38 -0.18 1.00 0.35 0.98 -0.05 -0.14 -0.00 0.07 -0.28 -0.12 -0.32 -0.09 0.20 0.24 -0.28
mouse_var -0.21 -0.33 -0.32 -0.21 0.35 1.00 0.35 0.05 -0.03 0.14 0.07 -0.14 -0.01 -0.33 -0.02 0.08 0.12 -0.27
aim_error -0.22 -0.33 -0.38 -0.22 0.98 0.35 1.00 -0.07 -0.15 -0.03 0.06 -0.18 -0.04 -0.33 -0.17 0.24 0.24 -0.24
movement_presses 0.42 0.34 0.13 0.42 -0.05 0.05 -0.07 1.00 0.25 0.65 -0.70 0.28 -0.09 0.34 0.19 -0.05 0.06 0.42
direction_changes 0.27 0.28 0.23 0.27 -0.14 -0.03 -0.15 0.25 1.00 0.22 -0.13 0.10 0.14 0.28 0.05 -0.08 -0.22 0.19
distance_moved 0.32 0.25 0.01 0.32 -0.00 0.14 -0.03 0.65 0.22 1.00 0.09 0.26 -0.05 0.25 0.36 -0.14 -0.06 0.26
idle_ratio -0.24 -0.22 -0.16 -0.24 0.07 0.07 0.06 -0.70 -0.13 0.09 1.00 -0.13 0.07 -0.22 0.09 -0.07 -0.13 -0.30
proximity_time 0.21 0.31 0.28 0.21 -0.28 -0.14 -0.18 0.28 0.10 0.26 -0.13 1.00 0.53 0.31 -0.37 0.17 -0.20 0.71
damage_taken -0.09 0.01 0.15 -0.09 -0.12 -0.01 -0.04 -0.09 0.14 -0.05 0.07 0.53 1.00 0.01 -0.62 0.21 -0.23 0.22
damage_dealt 0.91 1.00 0.74 0.91 -0.32 -0.33 -0.33 0.34 0.28 0.25 -0.22 0.31 0.01 1.00 0.12 -0.07 -0.28 0.77
player_health 0.17 0.12 0.00 0.17 -0.09 -0.02 -0.17 0.19 0.05 0.36 0.09 -0.37 -0.62 0.12 1.00 -0.72 -0.10 -0.14
low_health_time -0.04 -0.07 -0.05 -0.04 0.20 0.08 0.24 -0.05 -0.08 -0.14 -0.07 0.17 0.21 -0.07 -0.72 1.00 0.14 0.07
survival_time -0.27 -0.28 -0.28 -0.27 0.24 0.12 0.24 0.06 -0.22 -0.06 -0.13 -0.20 -0.23 -0.28 -0.10 0.14 1.00 -0.20
aggression_index 0.72 0.77 0.51 0.72 -0.28 -0.27 -0.24 0.42 0.19 0.26 -0.30 0.71 0.22 0.77 -0.14 0.07 -0.20 1.00
shots_fired shots_hit accuracy click_freq avg_mouse_speed mouse_var aim_error movement_presses direction_changes distance_moved idle_ratio proximity_time damage_taken damage_dealt player_health low_health_time survival_time aggression_index
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

ML METHODOLOGY

Models Explored:
- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest (Final Model)
- XGBoost
- Light GBM

Selected Model: Random Forest
- Achieved the best performance and stability on our dataset
- Handles non-linear relationships in gameplay behavior effectively
- Robust to noise and feature variability
- Performs relatively well with medium-sized datasets (~2200 samples)

How Random Forest Works:
- Ensemble of multiple decision trees
- Each tree is trained on a random subset of data and features
- Final prediction is made using majority voting

Input Features → Multiple Trees → Aggregated Prediction

Our hyperparameters: n-estimators = 300, class weight = ‘Balanced’

Why It Fits Our Problem:
- Gameplay behavior is complex and non-linear
- Different features contribute differently across scenarios
- Random Forest captures interaction between multiple behavioral signals

CHALLENGES

Key Challenges:

1. Limited Dataset Size
- ~2200 samples only
- Insufficient to fully capture variability in player behavior

2. Overlapping Affective States
- Similar behavioral patterns across states (e.g., Enjoying vs Frustration)
- Made classification difficult

How We Addressed These Challenges:

- Focused on collecting more gameplay data continuously
- Used models like Random Forest that are robust to noise

Key Insight:

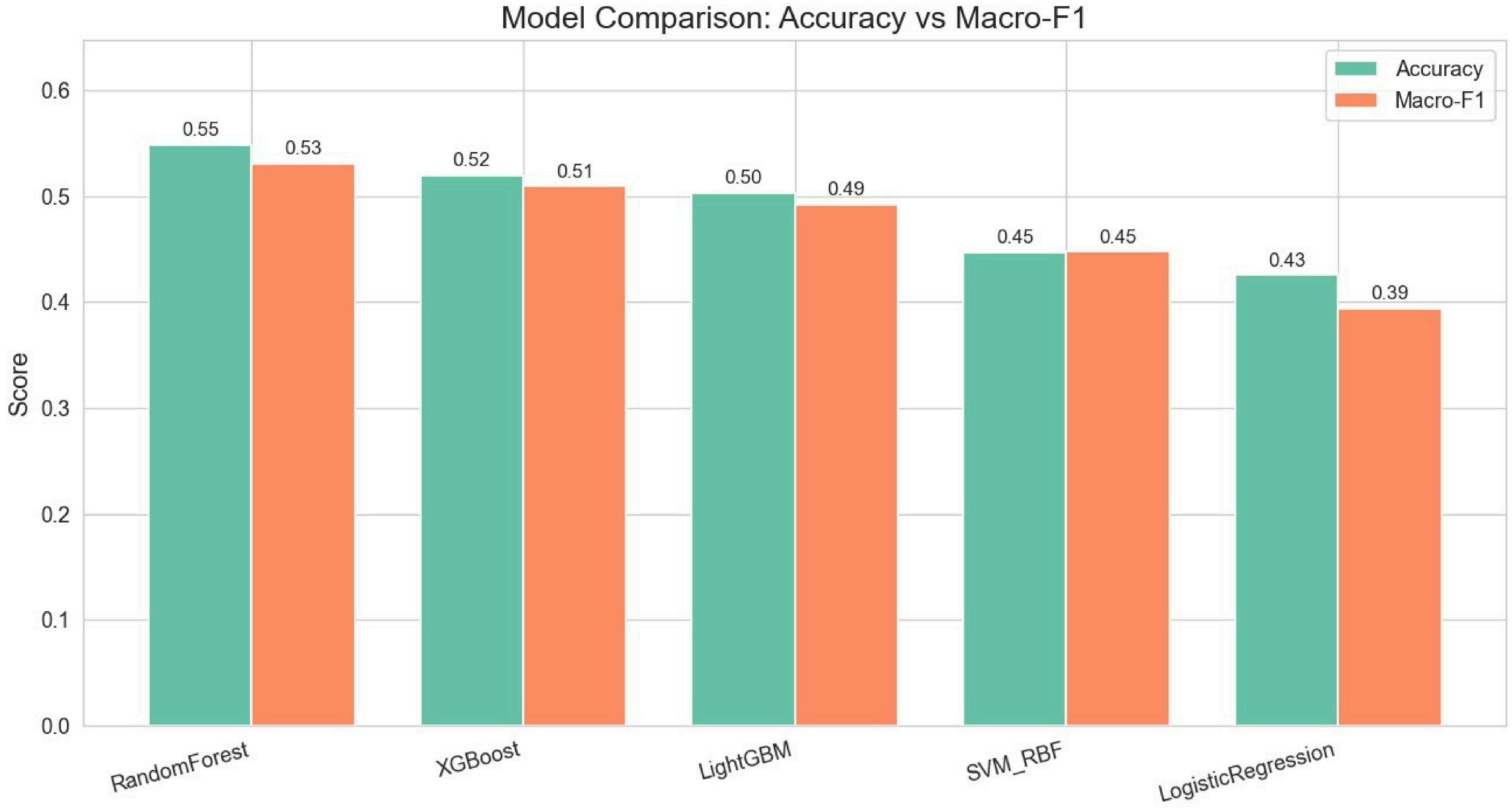
The primary bottleneck of our system is dataset size, and improving data quality and quantity is critical for better model performance.

PERFORMANCE METRICS

```
================================================================================
MODEL COMPARISON (4-Class, sorted by macro-F1)
================================================================================
             model  accuracy_mean  accuracy_std  macro_f1_mean  macro_f1_std  weighted_f1_mean  weighted_f1_std
      RandomForest         0.5483        0.0205         0.5303        0.0296            0.5413           0.0249
           XGBoost         0.5194        0.0094         0.5093        0.0160            0.5162           0.0114
          LightGBM         0.5033        0.0121         0.4919        0.0182            0.4996           0.0138
           SVM_RBF         0.4469        0.0269         0.4475        0.0270            0.4425           0.0302
LogisticRegression         0.4252        0.0204         0.3936        0.0208            0.4132           0.0189

Saved summary to model_comparison_results_updated.csv
Saved 5 confusion-matrix plots to plots_updated/

================================================================================
TOP MODEL (4-Class): RandomForest
================================================================================
Classification report (from cross_val_predict):
              precision    recall  f1-score   support

       Bored       0.56      0.48      0.51       404
    Enjoying       0.48      0.40      0.44       499
        Flow       0.61      0.51      0.56       451
  Frustrated       0.55      0.70      0.62       758

    accuracy                           0.55      2112
   macro avg       0.55      0.52      0.53      2112
weighted avg       0.55      0.55      0.54      2112
```

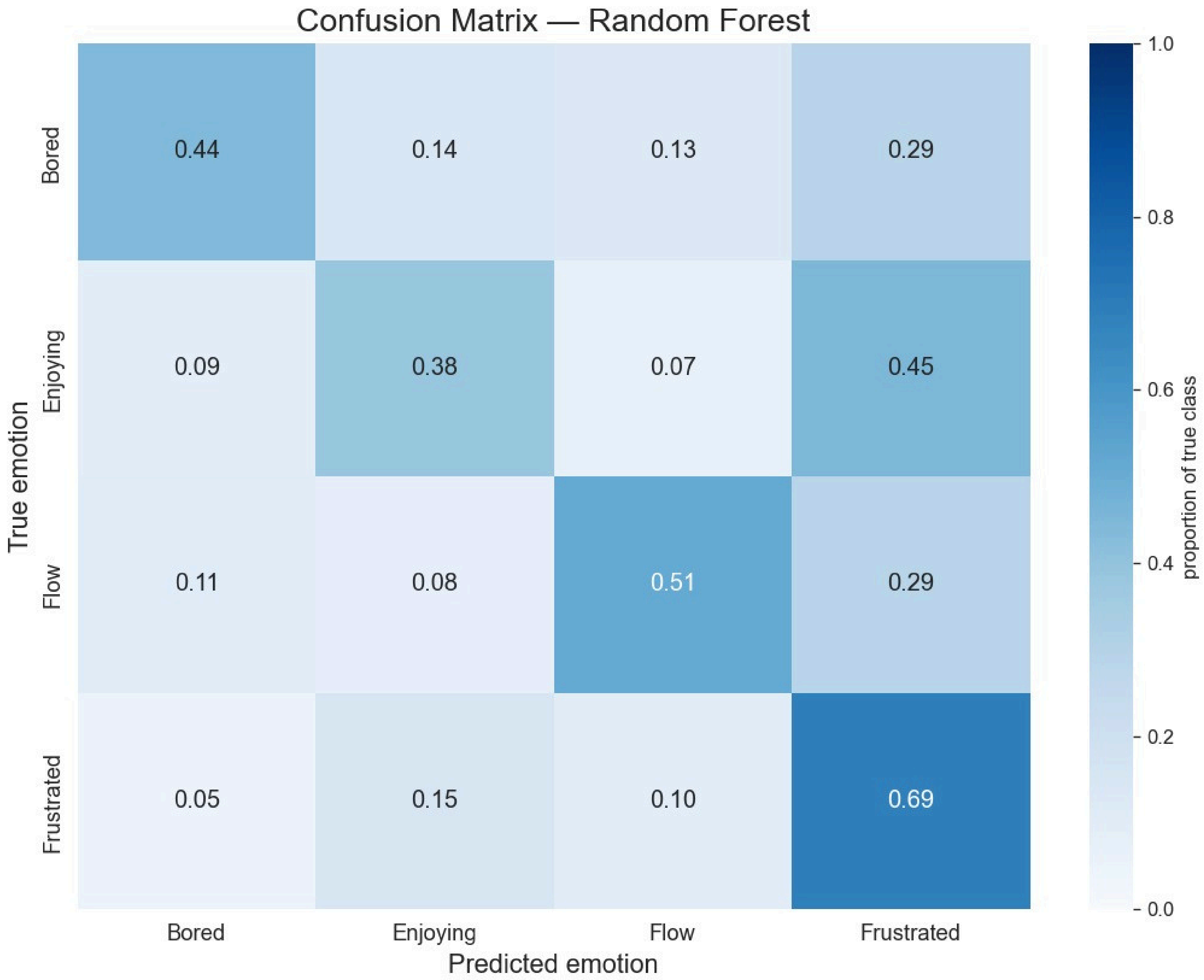
Highest Accuracy:
Random Forest
(54.8%)

PERFORMANCE METRICS



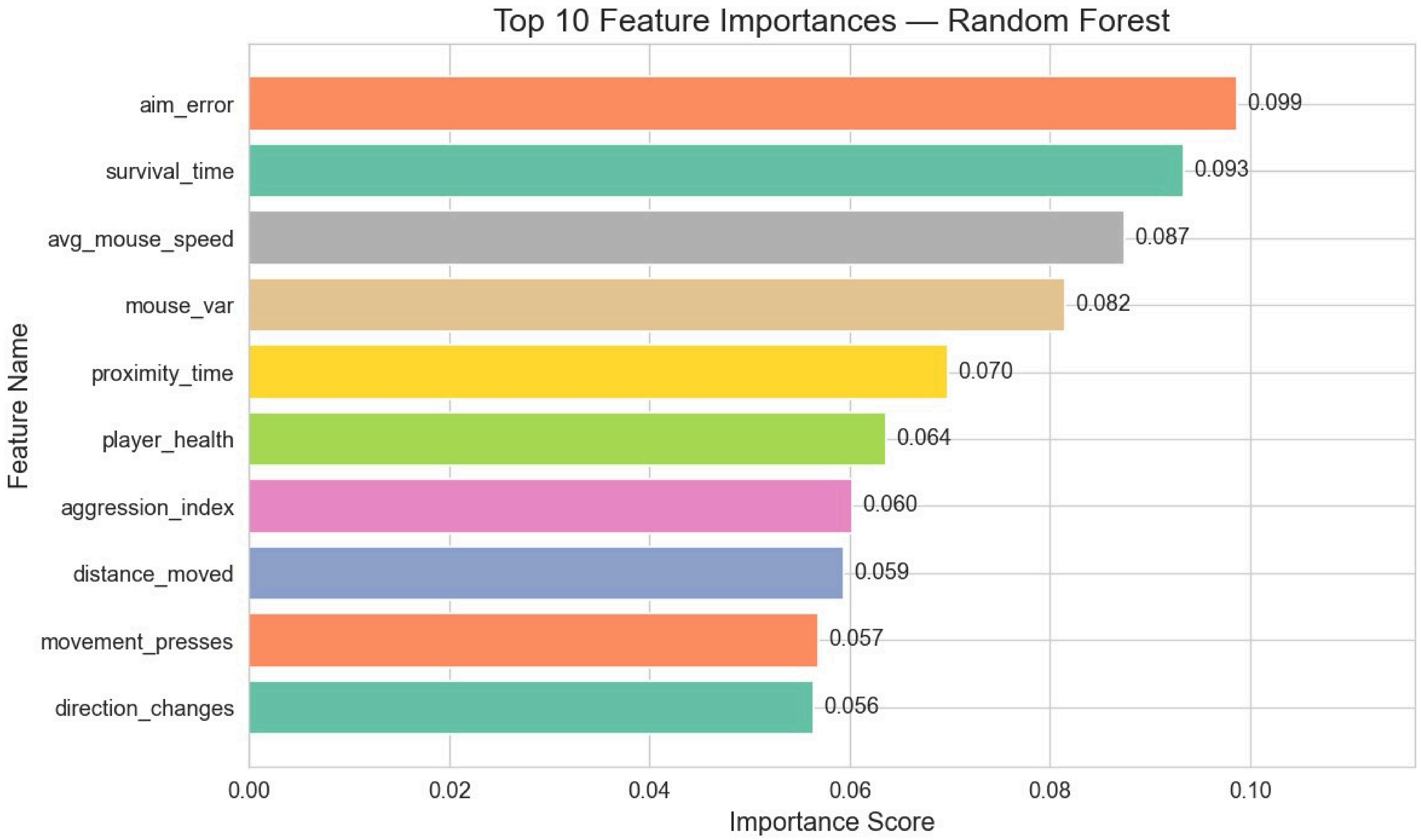
Random Forest achieved the highest Macro-F1 score (~0.53), indicating better balanced performance across all emotion classes compared to other models.

PERFORMANCE METRICS



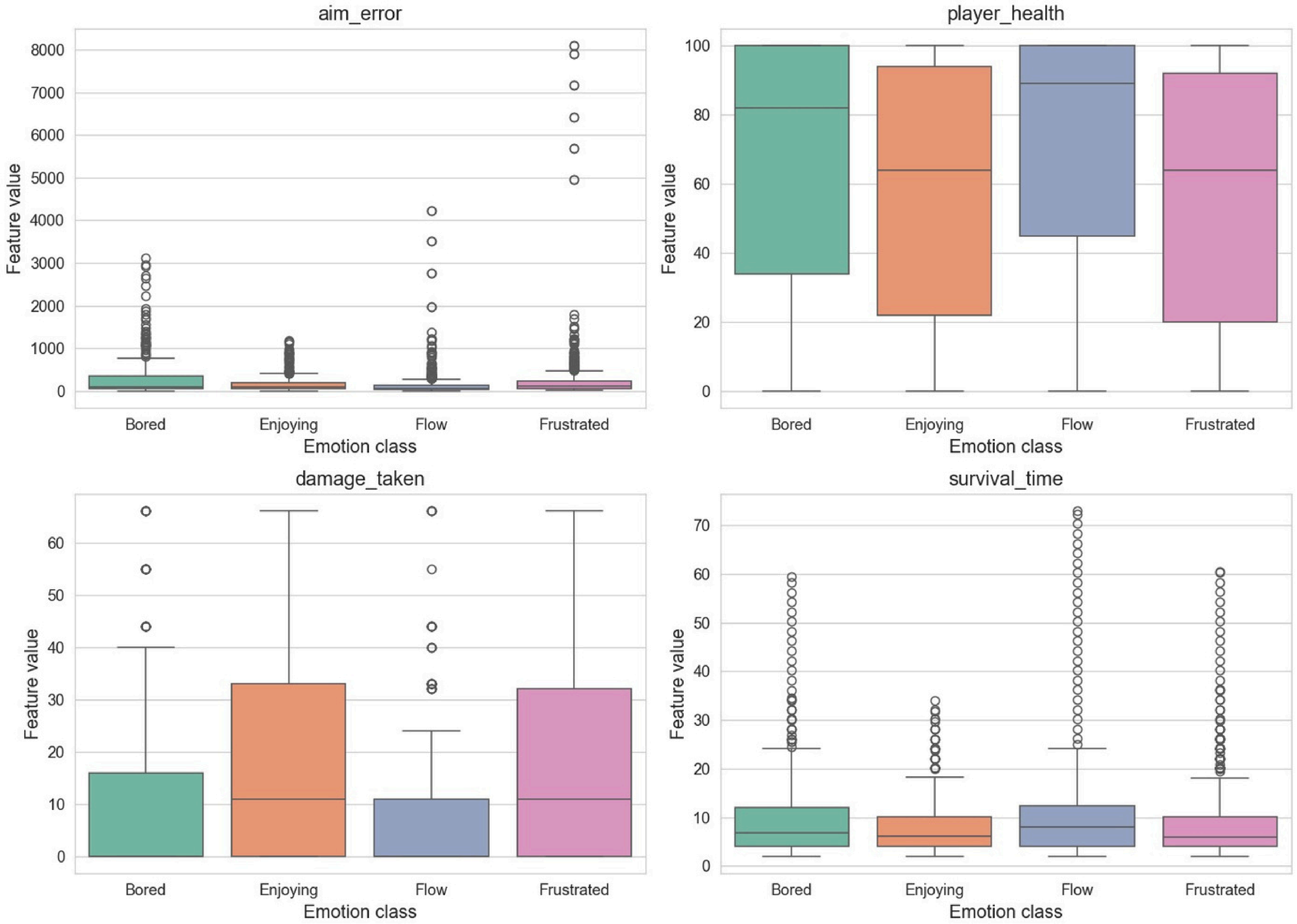
The model performs best on Frustrated (~0.69) and Flow (~0.51), but struggles with Enjoying and Bored. There is noticeable confusion between similar states, especially Enjoying and Frustrated.

MODEL INTERPRETATION

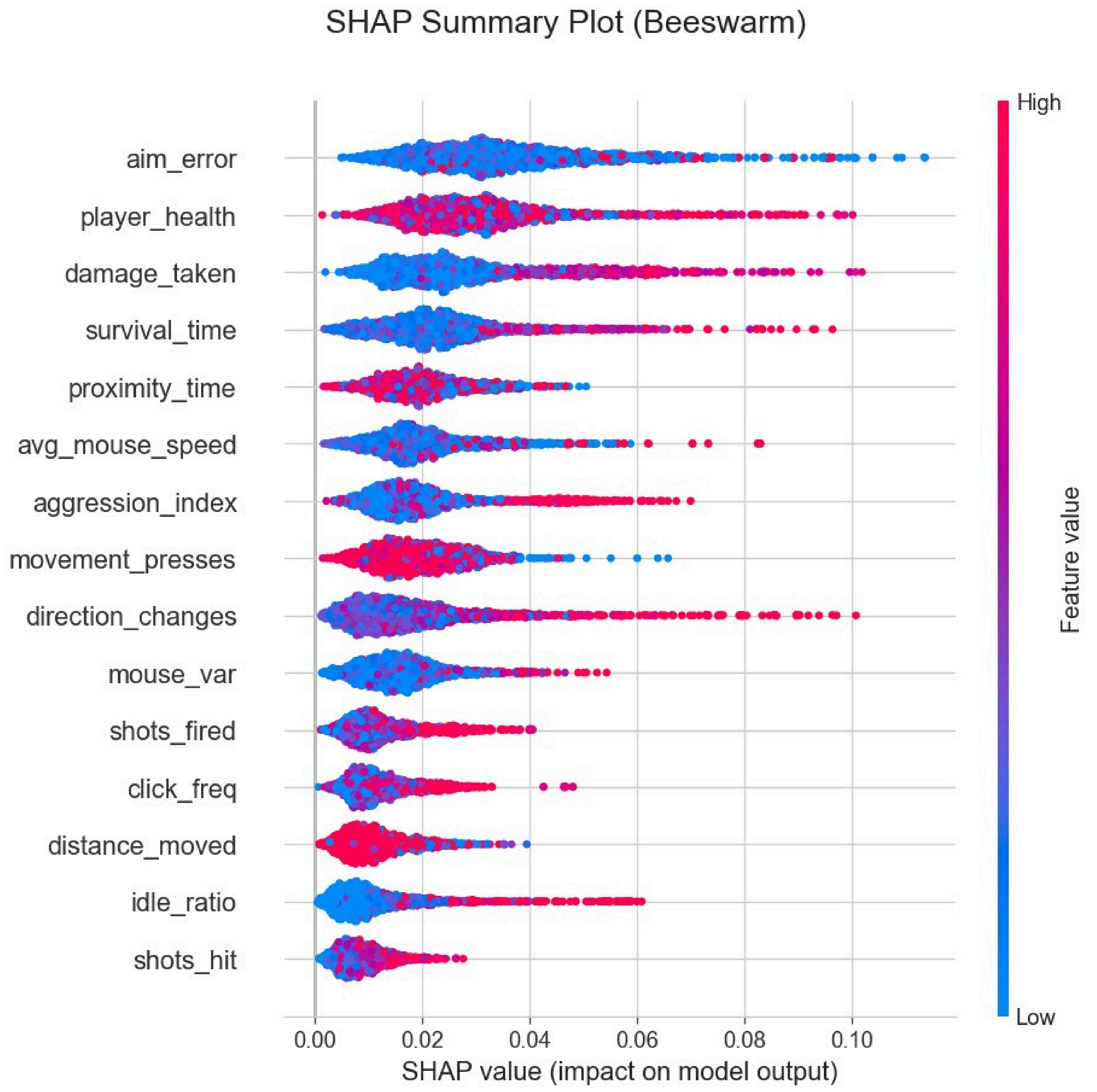


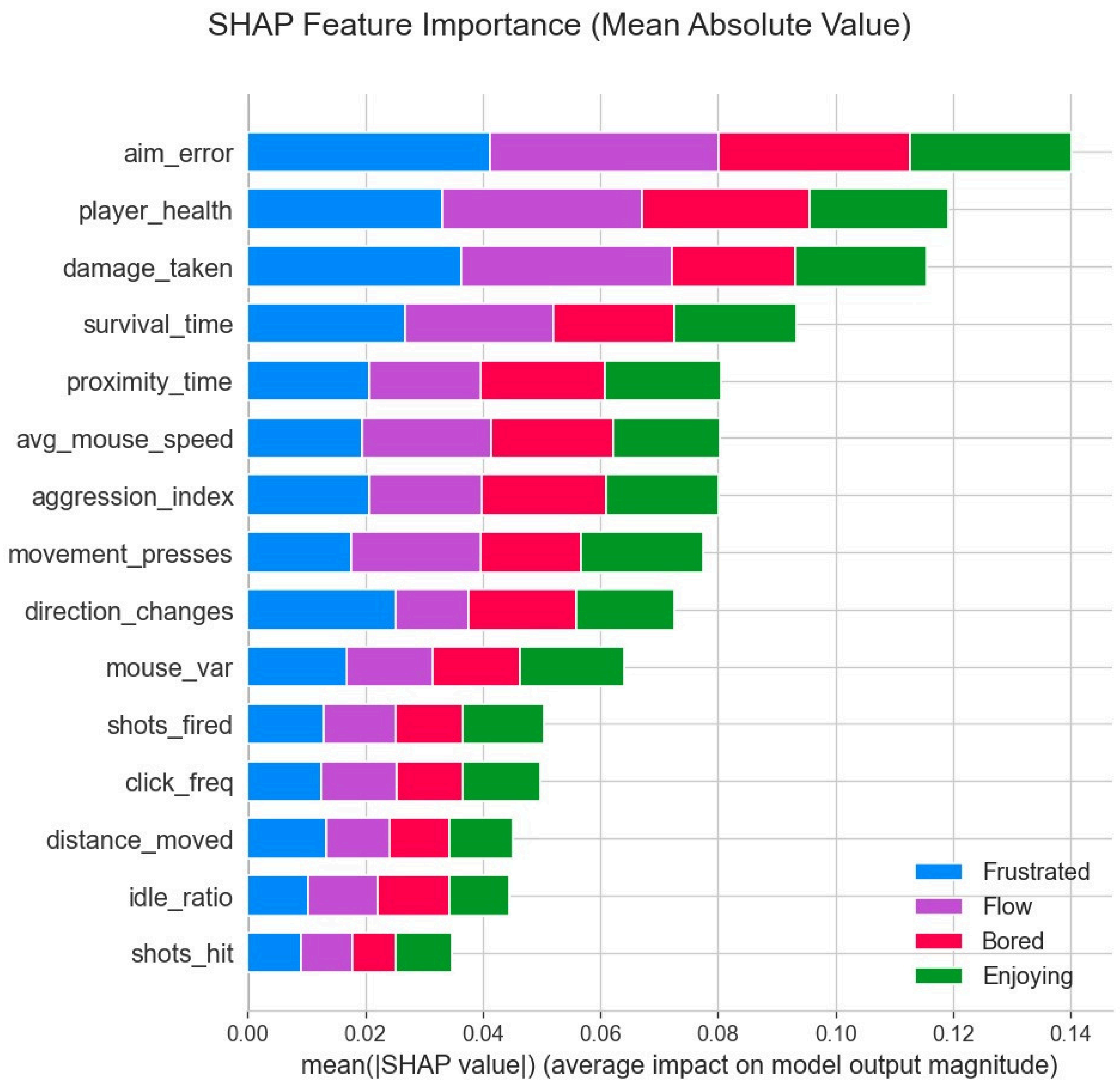
MODEL INTERPRETATION

Top Feature Distributions by Emotion Class



MODEL INTERPRETATION





DEPLOYABILITY

System Deployment

- Integrated trained model into the game for real-time affective state prediction
- Predictions generated using average of last 3 intervals using gameplay telemetry
- Difficulty dynamically adjusted based on predicted player state

Practical Advantages

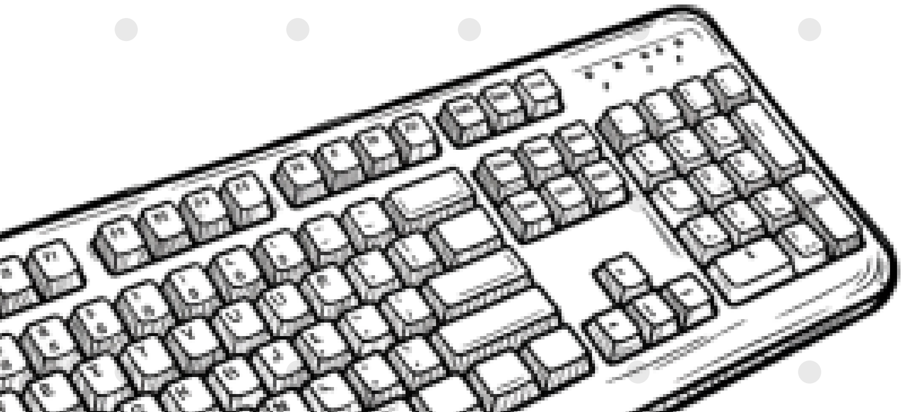
- Uses data from only keyboard and mouse
- Can be deployed in commercial games and interactive systems

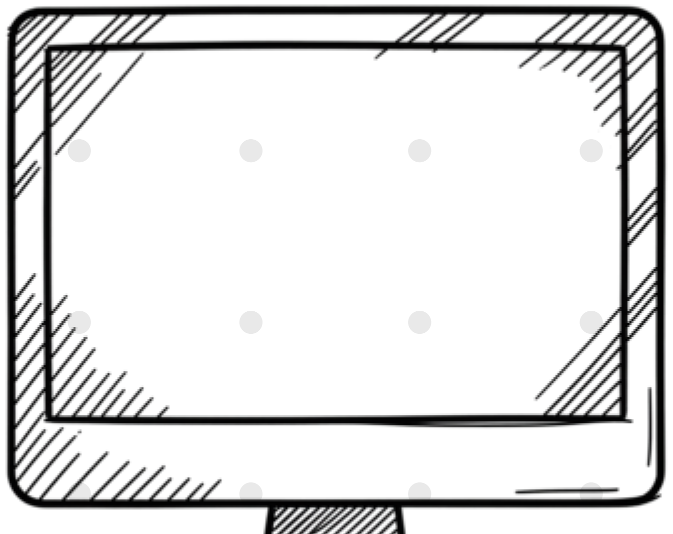
Current Limitation

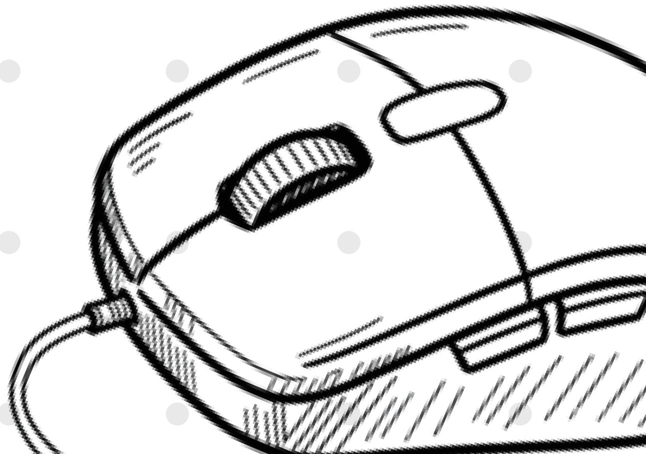
- No user study conducted to quantitatively evaluate improvement in engagement

Real-Time Pipeline

Gameplay → Feature Extraction → ML Prediction → Difficulty Adjustment



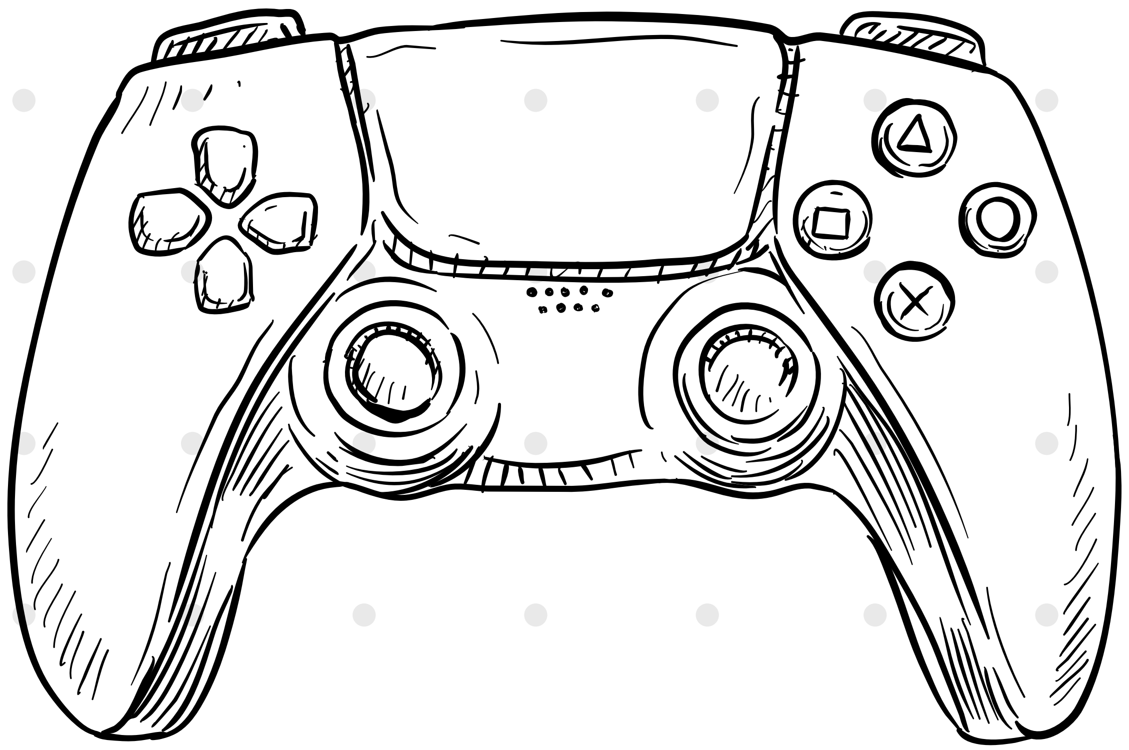




LIMITATIONS + FUTURE WORK

Evaluation and Validation

- Conduct user studies to measure:
    - engagement score
    - retention
    - player satisfaction
- Compare adaptive vs non-adaptive gameplay
- Increase dataset size



Challenges in Scaling:

Data Challenges
- Need large-scale, diverse datasets across different player types
- Increased variability in player behavior

Generalization Issues
- Model trained on one game may not perform well on other game genres
- Player behavior differs across contexts

Real-Time Constraints
- Low-latency prediction required for smooth gameplay
- Efficient feature computation needed at scale

Evaluation Challenges
- Measuring engagement objectively is difficult
- Requires controlled experiments and user feedback

Ethical Considerations
- Ensuring user consent and transparency
- Avoiding manipulation of player experience

MODEL-IMPLEMENTED GAME