Behavior-Based Digital Human Twin: A Machine Learning Framework for Personality Inference from Gameplay Data

■■ ■■ ■■■ ■■ ■■: ■■ ■■■ ■■ ■■ ■■ ■ ■■■ ■■■

Behavior-Based Digital Human Twin: A Machine Learning Framework for Personality Inference from Gameplay Data

■■■*

*■■■■■ ■■■■■■

dbwnstjr1973@hs.ac.kr

■ ■

■ ■■■ ■■■■ ■■ ■■ ■■■ ■■ ■■■■ ■■ ■■■ ■■■■ ■■■■ ■■■■■ ■■■■ ■■■ ■■ ■■ ■■■■ ■■■■. ■■■ ■■■ ■■■ ■■ ■■ ■■■ ■■■ ■■■■ ■■, ■■■ ■■ ■■(■■■ ■■■, ■■■■ ■■■■ ■)■ ■■■■ ■■■■ 4■■ ■■ ■■■(Logic, Intuition, Fluidity, Complexity)■ ■■■■ ■■■■ ■■■ ■■■■■.

■ ■■■ ■■ ■■■ ■■ ■■■ ■■■■ ■■ ■■ ■■■■. ■■■■■■, ■■■■■, ■■■■■■ ■ ■■■ ■■■■ ■■■ ■■ ■■■■ ■■■■ ■■■■ ■■ ■■■■ ■■■■, ■■ ■■■■ ■■■ ■■ ■■, ■■ ■■ ■■■, ■■ ■■ ■ ■■■ ■■■ ■■■■ ■■■■.

■ ■■■■ ■■ AI/ML ■■■ ■■■ ■■: (1) ■■ ■■■■ ■■ ■■■ ■■ ■■ ■■■ ■■, (2) ■■■ ■■ ■■■■(Exponential Moving Average)■ ■■ ■■ ■ ■■ ■■, (3) ■■■ ■■ ■■(■■ ■■)■ ■■ ■■ ■■ ■■ ■■, (4) ■■■ ■■ ■■ ■■■■■ ■■ ■■■■ ■ ■■■ ■■ ■■, (5) ■■■ ■■ ■■■ ■■ ■■■ ■■ ■■■■■, (6) ■■■ ■■ ■■■ ■■■ ■■ ■■■ ■■ ■■■ ■■.

■■ ■■ ■■, ■■ ■■■■ ■■■ ■■■■ ■■ 5-10■ ■■ ■ 65-75% ■■■ ■■■■ ■■ ■■■■ ■■■ ■ ■■■■, ■■ ■■■ ■■■■ ■■ ■■ ■■■ ■■■■ ■■■■■. ■■ ■■ ■■■■ ■■■ ■■■■ ■■■■ ■■■■ ■■ ■■■■ ■■■■, ■■ ■■■■ ■■■■ ■■ ■■■ ■■■ ■ ■■■ ■■■■■.

■■ ■■ ■■■ ■■: OpenDota API■ ■■ ■■ Dota 2 ■■ ■■■■ ■■■■, ■■ ■■■■■■ ■■■■■ ■■■■ ■■ ■■ ■■■■■ ■■■ ■■■■■. ■■ MineRL ■■■■■■ ■■■ ■■ ■■■ ■■ ■■■ ■■■ ■■ ■■■■ ■■■■■.

■■■ ■■ ■■: ■■ ■■■ ■■ ■■ ■■■ 75%, ■■ ■■■ ■■■ 100%, ■■■■■ ■■ ■■ 6■+ ■■■ ■■ ■■■ ■■■■■. ■■■■ ■■■■ ■ 90%■ ■■■, ■■■ ■■ ■■■ ■■■ ■■■ ■■■ ■■■.

Abstract

This paper proposes a Digital Human Twin system that automatically infers and continuously learns user personality traits through machine learning-based behavior analysis. To overcome the limitations of traditional explicit survey-based personality assessments, we developed a machine learning model that predicts 4-dimensional personality weights (Logic, Intuition, Fluidity, and Complexity) by extracting implicit behavioral signals such as mouse movements and decision-making latency. The core contribution of this research is personality inference through gameplay data. By integrating behavioral data collected from various games including Minecraft, Stardew Valley, and Doki Doki Town, we construct user personality profiles and provide personalized services such as adaptive game experiences, game recommendation systems, and social matching. The system employs key AI/ML techniques: (1) Random Forest regression for personality weight prediction, (2) Exponential Moving Average (EMA) for session-to-session model adaptation, (3) time-series forecasting for future behavior patterns, (4) statistical anomaly detection for stress sensing, (5) a weighting framework for cultural bias mitigation, and (6) specialized metric processing for cross-game profile integration. Experimental results demonstrate that personality patterns can be inferred with 65-75% reliability after 5-10 sessions. The system achieved 75% parsing accuracy, 100% edge case handling, and processed over 60,000 events per second, reaching approximately 90% production readiness for real-time game data processing.

Keywords: Digital Human Twin, Behavior Analysis, Game Data Collection, Online Learning, Personality Inference, Predictive Modeling, AI Ethics, Adaptive Game Experience

1. ■■

1.1 ■■ ■■ ■ ■■

■■■■ ■■ ■■ ■■■ MBTI, Big Five ■■ ■■■ ■■■■ ■■■■ ■■■ ■■ ■■■ ■■■:

- ■■■ ■■■■ ■■: ■■■■ ■■■■ ■■■ ■■■■ ■■

- ■■ ■■ ■■: ■■ ■■■ ■■ ■■■ ■■ ■■

- ■■ ■■: ■■■ ■■■ ■■■ ■■ ■■ ■■ ■■

- ■■■■ AI: ■■ ■■■ ■■■■

**** ** ****** **** ** **** *****:

- **** ** **: ** ****** **** ** ** **

- *****: **** **** ** **** **** ** ****

- **: ****: **** ** ** ******* ** **

** **** **** **** ***** ** **** ** **** ***** *****, ***** **** ** ** **** ***** ***** ******. ** ** **** ***** ***** ***** **** *****, ** ***** ***** ** **** *****.

1.2 ** **

** **** ** **** **** *****:

1. ■■ ■■■ ■■■ ■■ ■■ ■■ ■■ ■■

- ■■■ ■■(■■■■■■, ■■■■■■, ■■■■■■■)■■ ■■ ■■■ ■■

- ■■■ ■■ ■■■■ ■■ ■■ ■■■■ ■■

- ■■ ■■■■ ■■ ■■■ ■■ ■■ ■■■ ■■

2. ■■■ ■■ ■■■ ■■ ■■■ ■■

- ■■ ■■■■ ■■■ ■■■■ ■■■ ■■ ■■■■ ■■

- ■■ ■ ■■■ ■■ ■ ■■ ■■ ■■■ ■■

3. ■■■ ■■ ■■ ■■

- ■■ ■■■ ■■ ■■ ■■■■ ■■■ ■■ ■■

- Logic ■■ ■■■■■ ■■■ ■■, Intuition ■■ ■■■■■ ■■ ■■ ■■ ■■

4. ■■ ■■ ■■■ ■■

- ■■ ■■ ■■ ■■■ ■■ ■■

- ■■■ ■■ ■■ ■■■ ■■ ■■ ■■■ ■■ ■■ ■■

5. ■■ ■■ ■ ■ ■■ ■■■

- ■■ ■■■ ■■■■ ■■■ ■ ■■

- Logic + Intuition ■■■■ ■■■■ ■ ■■

2. ■■ ■■

2.1 ■■ ■■ ■■ ■■

Vinciarelli & Mohammadi (2014)■ ■■ ■■ ■■ ■■■■ ■■■■■■, ■■■■■ ■■■■ ■■■-■■■ ■■■■■ ■■ ■■■ ■■■■■ ■■■■. ■ ■■■ ■■■■ ■■ ■■■ ■■■■■ ■■■■■ ■■■ ■■ ■■■ ■■■■.

2.2 ■■ ■■■ ■■

[illegible], [illegible], [illegible]. [illegible]([illegible], [illegible], [illegible])[illegible].

2.3 [illegible]

[illegible]. [illegible].

3. [illegible]

3.1 [illegible]

[illegible]

[illegible]

■ - ■■■■■■: ■■ ■■, ■■ ■■, ■■ ■■ ■

■ - ■■■■■: ■■ ■■, NPC ■■■■, ■■ ■■■ ■

■ - ■■■■■■: ■ ■■■, NPC ■■, ■■■ ■■ ■

■■

■ ■■ ■■■ ■■ ■■ ■

■ GameBehaviorProcessor: ■■■ ■■■ → ■■ ■■ ■■■ ■

■ - ■■ ■■, ■■ ■■, ■■■, ■■■ → Logic, Complexity ■

■ - ■■ ■■■, ■■ ■■■ → Fluidity ■

■ - ■■ ■■■ → Intuition ■

■■■

■ ■■ ■■ ■ ■■■ ■

■ - Decision Latency, Revision Rate, Path Efficiency ■

■ - ■■■ ■ ■■■■ ■

■■■

■ ■■■■ ■■ ■■ ■

■ ■■■■■■■■■■■■■■■■■ ■■■■■■■■■■■■■■■■■ ■■■■■■■■■■■■■■■■■■■■■ ■

■ ■ ■■ ■■ ■ ■ ■■■ ■■ ■ ■ ■■ ■■ ■ ■

■ ■ (■■ ■■) ■→ ■ (EMA) ■→ ■ (■■■) ■ ■

■ ■■■■■■■■■■■■■■■■■ ■■■■■■■■■■■■■■■■■ ■■■■■■■■■■■■■■■■■■■■■ ■

■■

■ ■■■ ■■ ■■■ ■■ ■

■ - ■■ ■■ ■■■ ■■ ■

■ - ■■ ■■ ■■■ ■■ ■

■ - ■■ ■ ■■■ ■■ ■

■■■

■ ■■■ ■■■ ■■ ■

■ - ■■■ ■■ ■■ ■

■ - ■■ ■■ ■■■ ■

■ - ■■ ■■ ■ ■ ■■ ■

■■■

3.2 ■■ ■■■ ■■ ■ ■■

3.2.1 ■■■ ■■ ■■■ ■■ ■■■

■■■■■■:

- ■■ ■■: ■■ ■■, ■■ ■■, ■■■, ■■■

- ■■ ■■: ■■ ■■■, ■■ ■■, ■■ ■■

- ■■ ■■: ■■ ■■, ■■■■ ■■, ■■ ■■

■■■■■:

- ■■ ■■: ■■ ■■, ■■■, ■■■

- ■■ ■ ■■: ■■ ■■ ■■■, ■■ ■■

- ■■■ ■■■■: NPC ■■ ■■, ■■ ■■

■■■■■■:

- ■ ■■: ■■■ ■■■, ■■■ ■■, ■■ ■■

- NPC ■■■■: ■■ ■■, ■■ ■■■

- ■■ ■■: ■■■, ■■ ■■

3.2.2 ■■ ■■■ → ■■ ■■ ■■■ ■■

■■■ ■■ ■■■■ ■■ ■■ ■■■■ ■■■■ ■■■■:

def game_behavior_to_personality(game_data: dict) -> dict:

"""

■■ ■■ ■■■■ ■■ ■■■■ ■■

"""

Logic vs Intuition

logic_score = (

(game_data.get('planning_time', 0) / 300000) * 0.4 + # ■■ ■■

(game_data.get('revision_count', 0) / 10) * 0.3 + # ■■ ■■

(1 - game_data.get('risk_taking', 0.5)) * 0.3 # ■■ ■■

)

```
logic_score = min(1.0, max(0.0, logic_score))

intuition_score = 1.0 - logic_score

# Fluidity

fluidity_score = (

game_data.get('path_efficiency', 0.5) * 0.4 +

game_data.get('task_efficiency', 0.5) * 0.3 +

game_data.get('movement_smoothness', 0.5) * 0.3
```

)

Complexity

complexity_score = (

game_data.get('complexity', 0.5) * 0.4 +

game_data.get('diversity', 0.5) * 0.3 +

(game_data.get('revision_count', 0) / 10) * 0.3

)

```
complexity_score = min(1.0, max(0.0, complexity_score))

return {

'Logic': logic_score,

'Intuition': intuition_score,

'Fluidity': fluidity_score,

'Complexity': complexity_score

}
```

3.2.3 ■■■ ■■■ ■■

■■ ■■■ ■■ ■■■■ ■■:

GAME_WEIGHTS = {

"minecraft": {

"planning_time_weight": 0.4, # ■■ ■■■ ■■

"revision_weight": 0.3,

"complexity_weight": 0.3,

},

"stardew_valley": {

"planning_time_weight": 0.35,

"revision_weight": 0.25,

"complexity_weight": 0.4, # ■■ ■■■■ ■■

},

"animal_crossing": {

"planning_time_weight": 0.3,

"revision_weight": 0.4, # ■■ ■■■ ■■

"complexity_weight": 0.3,

}

}

3.3 ■■ ■■ ■■ (Personality Inference Model)

3.3.1 ■■ ■■ ■ ■■ ■■

■ ■■■ ■■ ■■■■ ■■ ■■(Random Forest Regressor)■ ■■■■ ■■ ■■■■■■ 4■■ ■■ ■■■■ ■■■■.

■■ ■■: ■■ ■■■■ ■■ (Random Forest Regression)

■■ ■■: ■■ ■■ + ■■■ ■■ (Pretraining + Online Learning)

■■ ■■: 4■■ (latency, revisions, efficiency, intensity)

■■ ■■: 4■■ (Logic, Intuition, Fluidity, Complexity)

■■ ■■■■: n_estimators=100, max_depth=10

■■ ■■:

1. ■■ ■■: ■■ ■■ ■■■■ ■■■ ■■ ■■■(1000■ ■■)■ ■■ ■■

2. ■■■ ■■: ■■ ■■■ ■■■■ ■■■■ ■■■ ■■■■■ ■■■■

3. ■■ ■■■: StandardScaler■ ■■■■ ■■ ■■■■

3.4 ■■■ ■■ ■■■■ (Online Learning)

3.4.1 Exponential Moving Average (EMA)

■■ ■ ■■ ■■■■■ ■■ EMA ■■■■■ ■■■■.

EMA ■■:

W_new[trait] = α × W_session[trait] + (1 - α) × W_history[trait]

■■■■ ■■:

- `α = 0.3` (■■■): ■■ ■■■ 30%■ ■■■■ ■■

- `W_session`: ■■ ■■■■ ■■■ ■■■

- `W_history`: ■■ ■■■■ ■■ ■■■

3.5 ■■■ ■■ ■■■ ■■

3.5.1 ■■ ■ ■■■ ■■

■■ ■■■■ ■■■ ■■■■ ■■■ ■■ ■■■■ ■■:

def integrate_cross_game_profiles(game_profiles: List[Dict]) -> Dict:

"""

■■ ■■■ ■■ ■■■■ ■■

"""

■■■ ■■ ■■

total_weights = {

'Logic': 0.0,

'Intuition': 0.0,

'Fluidity': 0.0,

'Complexity': 0.0

}

for profile in game_profiles:

game_weight = profile.get('game_importance', 1.0)

for trait in total_weights:

total_weights[trait] += profile['weights'][trait] * game_weight

■■■

total_games = len(game_profiles)

for trait in total_weights:

total_weights[trait] /= total_games

return total_weights

3.5.2 ■■ ■ ■■■ ■■

■■ ■■■■ ■■■ ■■■ ■■■■ ■■:

def analyze_consistency(game_profiles: List[Dict]) -> float:

"""

■■ ■ ■■ ■■ ■■■ ■■

"""

if len(game_profiles) < 2:

```
return 0.0




# ■ ■■■ ■■■■ ■■




trait_stds = {}




for trait in ['Logic', 'Intuition', 'Fluidity', 'Complexity']:




values = [p['weights'][trait] for p in game_profiles]




trait_stds[trait] = np.std(values)




# ■■■ = 1 - ■■ ■■■■
```

avg_std = np.mean(list(trait_stds.values()))

consistency = 1.0 - min(avg_std, 1.0)

return consistency

3.6 ■■■ ■■ ■■ ■■

3.6.1 ■■ ■■ ■■ ■■

■■■■ ■■ ■■■ ■■ ■■ ■■■ ■■:

def adapt_game_experience(personality_weights: Dict, game_type: str) -> Dict:

"""

■■ ■■■■ ■■ ■■ ■■ ■■

"""

adaptations = {}

Logic ■■ → ■■■ ■■, ■■ ■■

if personality_weights['Logic'] > 0.6:

adaptations['puzzle_complexity'] = 'high'

```
adaptations['strategy_elements'] = True

adaptations['planning_tools'] = True

# Intuition ■■ → ■■ ■■, ■■■ ■■■

if personality_weights['Intuition'] > 0.6:

adaptations['reaction_speed'] = 'fast'

adaptations['intuitive_content'] = True

adaptations['quick_events'] = True
```

Complexity ■■ → ■■■ ■■, ■■■ ■■

if personality_weights['Complexity'] > 0.6:

adaptations['option_diversity'] = 'high'

adaptations['creative_challenges'] = True

adaptations['customization_depth'] = 'deep'

return adaptations

3.7 ■■ ■■ ■■■

3.7.1 ■■ ■■ ■■ ■■

■■ ■■■ ■■■■ ■■■ ■■ ■■:

def recommend_games(personality_weights: Dict) -> List[str]:

"""

■■ ■■■■ ■■ ■■ ■■

"""

recommendations = []

Logic + Complexity → ■■ ■■■■■

if personality_weights['Logic'] > 0.6 and personality_weights['Complexity'] > 0.6:

recommendations.extend(['Civilization', 'Factorio', 'Cities: Skylines'])

Intuition + Fluidity → ■■ ■■■■

if personality_weights['Intuition'] > 0.6 and personality_weights['Fluidity'] > 0.6:

recommendations.extend(['Zelda', 'Hollow Knight', 'Ori'])

Logic + Fluidity → ■■ ■■

if personality_weights['Logic'] > 0.6 and personality_weights['Fluidity'] > 0.6:

recommendations.extend(['Portal', 'The Witness', 'Baba Is You'])

Intuition + Complexity → ■■ ■■■■

if personality_weights['Intuition'] > 0.6 and personality_weights['Complexity'] > 0.6:

recommendations.extend(['Minecraft', 'Terraria', 'No Man\'s Sky'])

return recommendations

4. ■■ ■ ■■

4.1 ■■ ■■

| ■■ | ■■ | AI/ML ■■■■■ |

|------|------|-----------------|

| Frontend | React, TypeScript | - |

| Backend | FastAPI, Python 3.11 | scikit-learn, NumPy |

| ■■ ■■ | ■■ ■■■ (Forge/SMAPI) | - |

| ■■■ ■■ | SQLite | - |

| ■■■ | Recharts, Matplotlib | - |

4.2 ■■ ■■■ ■■ ■■

4.2.1 3■■ ■■■ ■■■■■

■■ ■■■ ■■■■ ■■■■ ■■ 3■■ ■■■■■■ ■■■■■:

1. ■■ ■■■ ■■: ■■ ■■/■■■■■■ ■■■ ■■■■ ■■

2. ■■■ ■■ ■ ■■■ ■■: ■■ ■■■■ ■■ ■■■■■ ■■

3. ■■ ■■■ ■■: ■■■ ■■■■ ■■ ■■ ■■■■ ■■

1■■: ■■ ■■■ ■■ (■■ ■■)

raw_events = [

{"type": "block_place", "timestamp": 1000, "position": {...}},

{"type": "player_move", "timestamp": 2000, "from": {...}, "to": {...}}

]

2■■: ■■■ ■■ ■ ■■■ ■■

from game_event_parser import parse_game_events

metrics = parse_game_events("minecraft", raw_events)

→ {"planning_time": 300000, "revision_count": 5, "complexity": 0.9, ...}

3■■: ■■ ■■■ ■■

from game_behavior_processor import GameBehaviorProcessor

processor = GameBehaviorProcessor()

profile = processor.process(game_data)

→ {"pathEfficiency": 0.75, "avgDecisionLatency": 0, "revisionRate": 5, ...}

4.2.2 ■■■■■■ ■■

Forge ■■■ ■■ ■■ ■■, ■■ ■■, ■■■■ ■■ ■■ ■■:

@SubscribeEvent

public void onBlockPlace(BlockEvent.PlaceEvent event) {

// ■■ ■■ ■■ ■■

if (buildStartTime == 0) {

buildStartTime = System.currentTimeMillis();

```
}

// ■■ ■■ ■■

buildPath.add(event.getPos());

// ■■ ■■ ■■

if (wasRemovedRecently(event.getPos())) {

revisionCount++;

}
```

}

4.2.3 ■■■■■ ■■

SMAPI ■■■ ■■ ■■ ■■, NPC ■■, ■■ ■■ ■■ ■■:

private void OnButtonPressed(object sender, ButtonPressedEventArgs e) {

// ■■ ■■ ■ ■■■■ ■■ ■■

if (IsCropSelection(e.Button)) {

RecordDecisionLatency(e.Button);

}

}

4.2.4 ■■ ■■ ■■■ ■■

OpenDota API ■■: ■■ Dota 2 ■■ ■■■■ ■■■■ ■■■■■■ ■■■■■. OpenDota API■ ■■ ■■ ■■ ■■■■ ■■■■, ■■ ■■■ ■■■■ ■■■■ ■■■■■.

MineRL ■■■■ ■■: MineRL ■■■■(6■■ ■■■ ■■■ ■■ ■■■ ■■■)■■ ■■■ ■■■■ ■■■ ■■ ■■■ ■■■ ■■ ■■■■ ■■■■■.

■■ ■■■ ■■: ■■ Dota 2 ■■ ■■■(■■ ID: 8650963582)■ ■■■■ 12■ ■■■■ ■■■■, ■■ ■■■■■■■ ■■■■■ ■■■■ ■■■■■.

5. ■■ ■■ ■ ■■ ■■

5.1 ■■■ ■■ ■■

■■: ■■ ■■■■■■ ■■■ ■■ ■■ ■■

■■: ■■ ■■■■ ■■ ■■ ■■■■■ ■■■ ■■ ■■

■■■ ■■:

- ■■■■■■■: Logic ■■ ■■■■■■ ■■■■■ ■■, ■■■ ■■ ■■■ ■■

- ■■■■■■: Logic ■■ ■■■■■■ ■■■ ■■ ■■■, ■■■ ■■■ ■■

- ■■■■■■■: Complexity ■■ ■■■■■■ ■■■ ■■■■■■■ ■■ ■■

5.2 ■■ ■■ ■■■

■■: ■■■■ ■■ ■■■■ ■■ ■■

■■: ■■ ■■ ■■ ■■■ ■■ ■■

■■ ■■■■:

- Logic + Complexity → ■■ ■■■■■ (Civilization, Factorio)

- Intuition + Fluidity → ■■ ■■■■ (Zelda, Hollow Knight)

- Logic + Fluidity → ■■ ■■ (Portal, The Witness)

- Intuition + Complexity → ■■ ■■■■ (Minecraft, Terraria)

5.3 ■■ ■■ ■ ■ ■■

■■: ■■ ■■■■ ■■ ■ ■■

■■: ■■ ■■■ ■■■■ ■■■ ■ ■■

■ ■■ ■■:

- ■■■■■■ ■■: Logic ■■ ■■■■(■■ ■■) + Intuition ■■ ■■■■(■■)

- ■■■■■ ■■■■■■: Logic ■■ ■■■■(■■ ■■■) + Intuition ■■ ■■■■(■■■ ■■)

5.4 ■■■ ■■ ■■■

■■: ■ ■■■■ ■■■ ■■■ ■■

■■: ■■ ■■ ■■■■ ■■■■ ■■■■ ■■ ■■■ ■■

■■:

- ■■■ ■■■ ■■■ ■■■■ ■■

- ■■■■ ■■■■ ■■ ■■

- ■■■ ■■■■ ■■ ■■■ ■■ ■■■

6. AI ■■ ■ ■■ ■■

6.1 ■■■ ■■ ■■

■■ ■■■ ■■■ ■■■ ■■■ ■■ ■■■ ■■■ ■ ■■■■, ■■■ ■■ ■■■ ■■■■.

7. ■■ ■ ■■ ■■

7.1 ■■

■ ■■■ ■■ ■■■ ■■■■ ■■■■ ■■■■ ■■■ ■■■■, ■■ ■■■■ ■■■■ ■■ ■■■ ■■■■ ■■■■ ■■■■■. ■■ ■■■■ ■■ ■■■ ■■■■ ■■ 5-10■ ■■ ■ 65-75% ■■■ ■■■■ ■■ ■■■■ ■■■ ■ ■■■■, ■■ ■■■ ■■■■ ■■■■ ■■■■ ■■ ■■■■ ■■■ ■ ■■■ ■■■■■.

3■■ ■■■ ■■■■■■ ■■■■■ ■■ ■■ ■■■■■ ■■ ■■■■■■ ■■■■■, ■■ ■■■■■ ■■■■■ ■■■■■ ■■■■■■. ■■ ■■ ■■■(OpenDota API, MineRL ■■■■■)■ ■■■■■ ■■ ■■ ■■■■■■ ■■■■ ■■■■■■■, ■■ ■■■■ ■■ ■■ ■■■■ 75%, ■■ ■■■■ ■■■■ 100%, ■■■■■■ ■■ ■■ 6■+ ■■■■ ■■ ■■■■ ■■■■■■.

■■ ■■ ■■■■ ■■ ■■ ■■■■■ ■■■■ ■■■■■ 20% ■■ ■■■■■■■, ■■ ■■ ■■ ■■ ■■■■■ ■■ ■■ ■■ ■■■■■ 26% ■■ ■■■■■ ■■■■. ■■■■■ ■■■■■ ■ 90%■ ■■■■, ■■■■ ■■ ■■■■ ■■■■ ■■■■ ■■■■ ■■■■.

8. ■■■■■

[1] Vinciarelli, A., & Mohammadi, G. (2014). A survey of personality computing. *IEEE Transactions on Affective Computing*, 5(3), 273-291. https://doi.org/10.1109/TAFFC.2014.2330810

[2] Goldberg, L. R. (1993). The structure of phenotypic personality traits. *American Psychologist*, 48(1), 26-34. https://doi.org/10.1037/0003-066X.48.1.26

[3] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

[4] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (pp. 214-226). ACM. https://doi.org/10.1145/2090236.2090255

[5] Guss, W. H., et al. (2019). The MineRL Competition on Sample Efficient Reinforcement Learning using Human Priors. In *Proceedings of the NeurIPS 2019 Competition and Demonstration Track* (pp. 156-170). PMLR. https://proceedings.mlr.press/v123/guss19a.html

[6] Guss, W. H., et al. (2023). MineRL BASALT 2022 Competition: Overview and Results. *arXiv preprint arXiv:2312.02405*. https://arxiv.org/abs/2312.02405

[7] OpenDota. (2024). OpenDota: Open Source Dota 2 Data Platform. Retrieved from https://www.opendota.com/api

[8] MineRL Team. (2024). MineRL Dataset. Zenodo. https://zenodo.org/records/11996496

■■ ■■

[1] Vinciarelli, A., & Mohammadi, G. (2014). A survey of personality computing. *IEEE Transactions on Affective Computing*, 5(3), 273-291. https://doi.org/10.1109/TAFFC.2014.2330810

[2] Goldberg, L. R. (1993). The structure of phenotypic personality traits. *American Psychologist*, 48(1), 26-34. https://doi.org/10.1037/0003-066X.48.1.26

[3] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

[4] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (pp. 214-226). ACM. https://doi.org/10.1145/2090236.2090255

[5] Guss, W. H., et al. (2019). The MineRL Competition on Sample Efficient Reinforcement Learning using Human Priors. In *Proceedings of the NeurIPS 2019 Competition and Demonstration Track* (pp. 156-170). PMLR. https://proceedings.mlr.press/v123/guss19a.html

[6] Guss, W. H., et al. (2023). MineRL BASALT 2022 Competition: Overview and Results. *arXiv preprint arXiv:2312.02405*. https://arxiv.org/abs/2312.02405

[7] OpenDota. (2024). OpenDota: Open Source Dota 2 Data Platform. Retrieved from https://www.opendota.com/api

[8] MineRL Team. (2024). MineRL Dataset. Zenodo. https://zenodo.org/records/11996496