Cheating Detection in Multiplayer Online Games

Kashif Nazir
knazir@stanford.edu

Overview

- **Motivation**: With the rise of competitive multiplayer online games (MOGS), cheating has become more prevalent due to new incentives.
- **Insight**: Cheating is considered anomalous relative to typical player behavior.
- **Solution**: To detect whether a player is cheating, we'll use two approaches in tandem:
  - An unsupervised model to infer typical player behavior and detect anomalies.
  - A supervised model to classify player behavior as cheating or not.

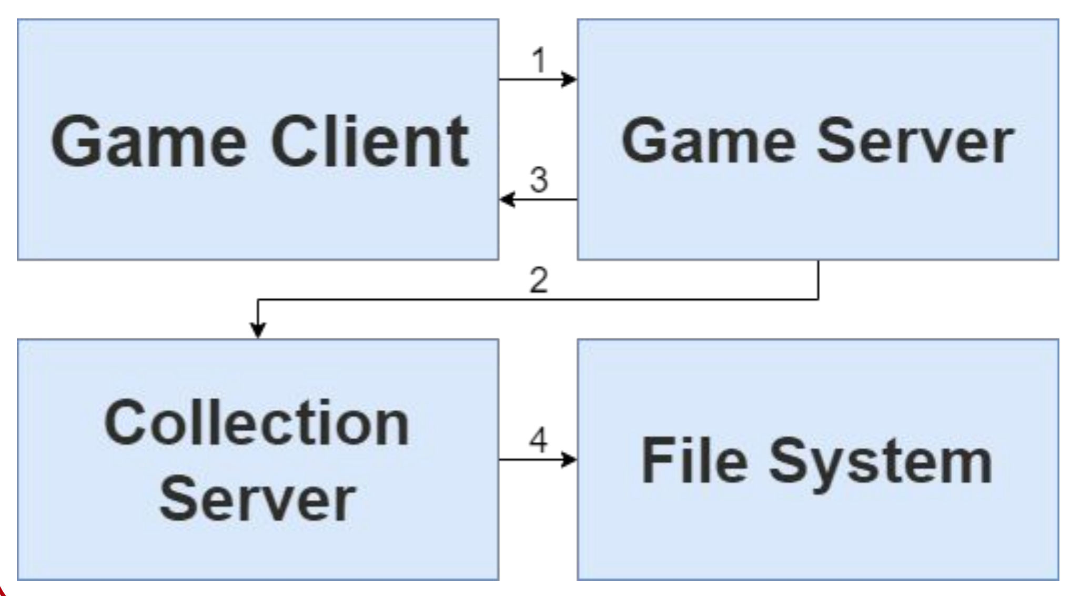
Representing Game Data

- The game server updates information about the game world at a specific **tick rate**.
- We represent a specific point in time of a game (a **tick**) as a **game state**.
- A game state is a dictionary containing team assignments, scores, positions, etc.

Data Processing

- **Game**: We modified an open-source first-person shooter MOG to collect >300K game states.
- **Cheats**: We implemented two cheats:
  - **Aimbot**: Automatically locks the player's aim on the nearest enemy.
  - **Rapid Fire**: Sends keystrokes in rapid succession to allow more actions.
- **Filtering**: We examined only the game states leading up to a player scoring a point.
- **Preprocessing**: We computed changes in player behavior in the moments before a point.
- **Features**: For each cheat, we collected:
  - **Aimbot**: Maximum change in aim yaw (horizontal angle) and pitch (vertical angle)
  - **Rapid Fire**: Number of keystrokes received and rate of damage dealt to the opponent

Data Collection

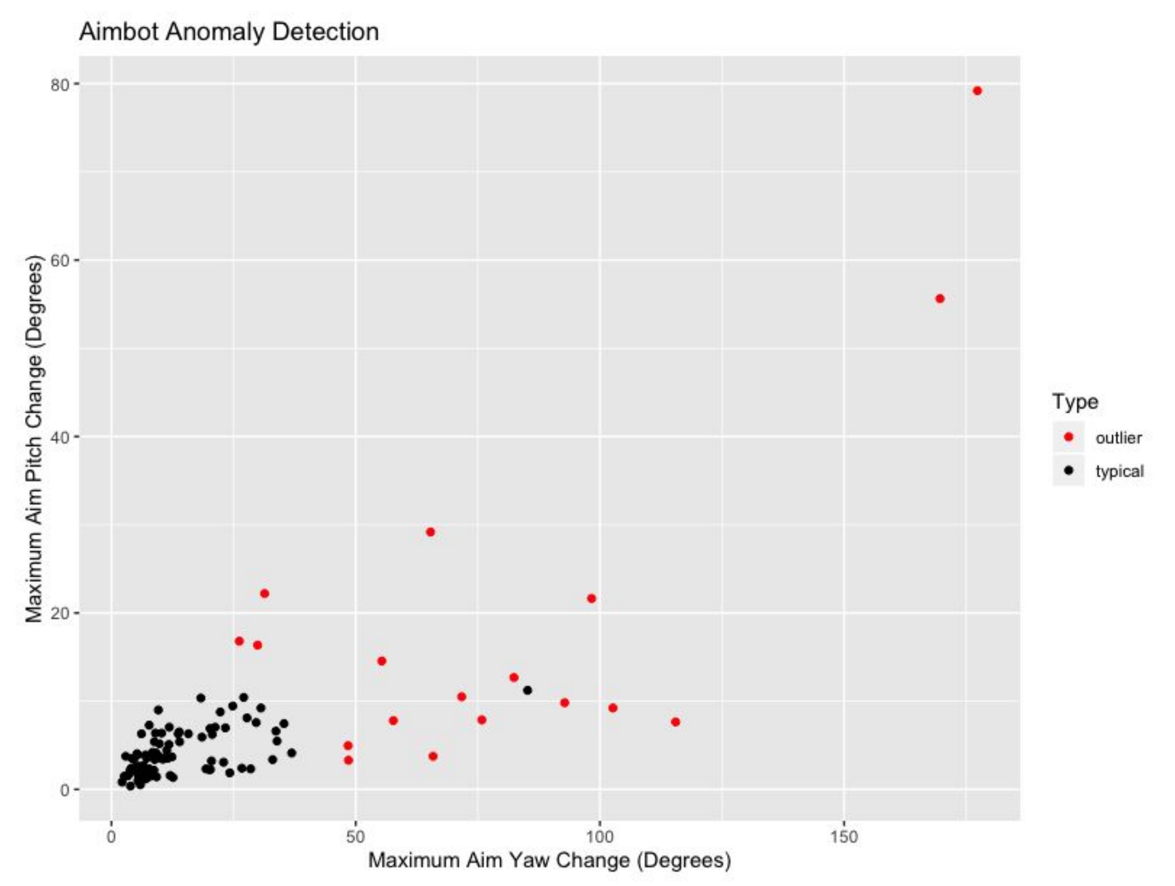


1. The player's game client sends its input and local game state to the game server.
2. The game server computes the new game state based on the input from all connected players (e.g. new player positions, health, etc.), then sends this new state to the collection server.
3. The game server sends back the updated game state to all connected players.
4. The collection server writes out the latest game state data received from the game server to the file system for storage.

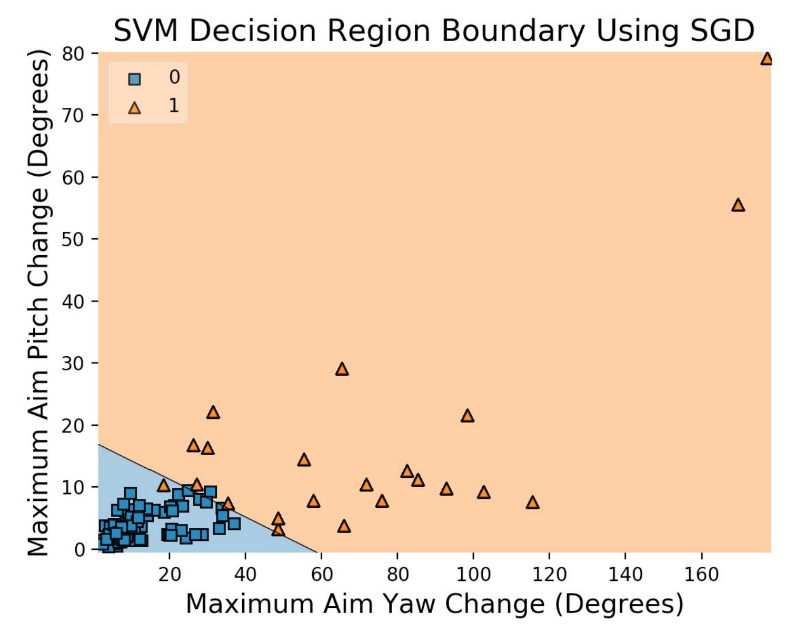
Anomaly Detection

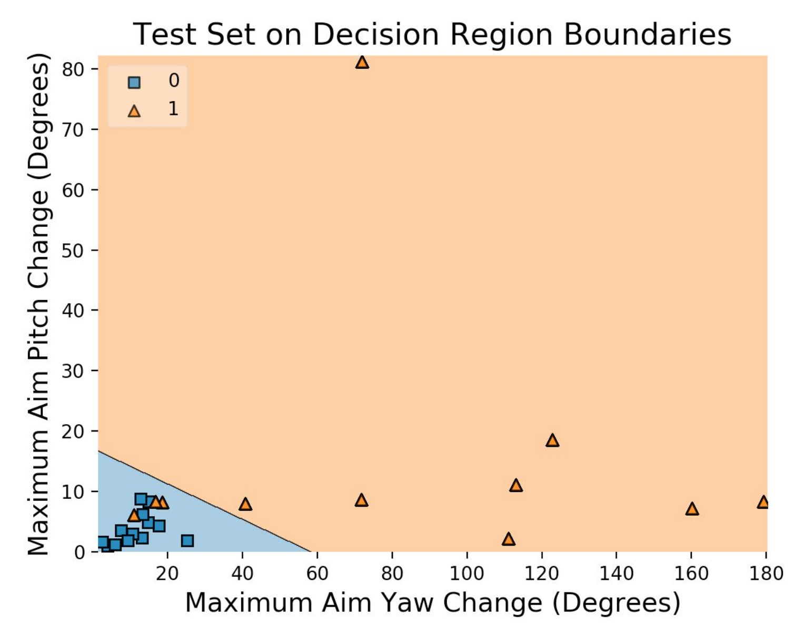
Our unsupervised model used an algorithm based on K-Nearest-Neighbors (KNN) to find anomalies in high-dimensional data. We effectively clustered and visualized player behavior.

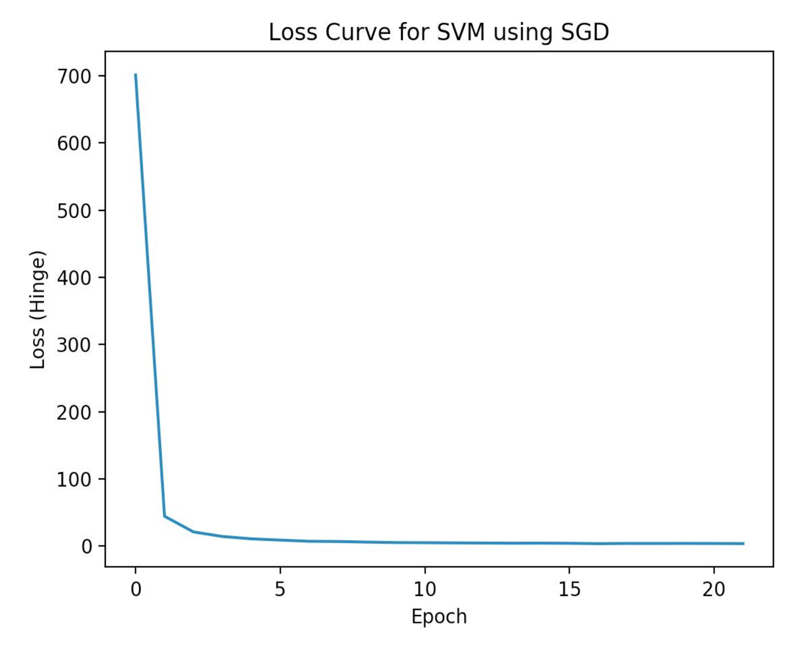
- Used **STRAY** (STReam AnomalY) algorithm to detect anomalous behavior based on KNN. [1]
- Fast nearest-neighbor search allowed us to stream new player behavior in real-time.
- Outliers taken as tail-end of nearest neighbors clusters with hyperparameters **α=0.2, k=10**.
- Effectively identified clear anomalies, but struggled with microclusters near fringes.



Classification







Our supervised model used an **SVM** trained using **SGD** and split with **75%/25%** train/test.

- SVM made decisions on individual observations of behavior changes.
- Classified each kill based on threshold of cheating behavior changes exceeding 0.5.
- Decision boundaries identified clearly separable behavior for aimbot detection.

Results

Anomaly Detection (based on known labels)

	Accuracy	Precision	Recall	F1
Aimbot	0.750	0.871	0.844	0.857
Rapid Fire	0.511	0.542	0.597	0.284

Cheating Classification

	Accuracy	Precision	Recall	F1
Aimbot	0.741	0.833	0.870	0.851
Rapid Fire	0.728	0.795	0.866	0.830

Discussion

- This approach effectively identified aimbot in both unsupervised and supervised settings.
- SVM was better than anomaly detection on rapid fire cheat due to placing cheating microclusters on the correct side of decision boundary independent of proximity to other observations.
- Effectiveness is heavily dependent on domain knowledge of high-signal features.
- Exhibited low false positive rate, important to not mislabel innocent players as cheating.
- Training on entire game state causes overfitting.

Future Work

- Combine SVM and threshold prediction into a single neural network.
- Immediately prune outliers to prevent micro-clustering of anomalies.
- Automate feature selection for different cheats.
- Explore detection in different genres of MOGs.
- Investigate general clustering of player behavior.

References

[1] Priyanga Dilini Talagala, Rob J. Hyndman, and Kate Smith-Miles. Anomaly detection in high dimensional data, 2019.