Win, Lose or Cheat: The Analytics of Player Behaviors in Online Games

Johanne Christensen,
Mark Cusick,
Andrea Villanes
North Carolina State University
{jtchrist, mbcusick,
avillan}@ncsu.edu

Oleg Veryovka
oleg@terraknowa.com

Ben Watson,
Michael Rappa
North Carolina State University
{bwatson,
mrappa}@ncsu.edu

ABSTRACT

Analysis of game telemetry data is fundamental for understanding common player behaviors and combating cheating in massively multi-player online games (MMO). The challenge of such analysis is data contamination resulted from system and network latency, data tempering and bugs, In this paper we are analyzing race results from a racing MMO. We are addressing the challenge of identifying true race winners and detecting cheaters. Our approach is based on statistical analysis techniques for detecting outliers. Outliers are classified using heuristics of multiple measures. We identify potential cheaters by analyzing outlier results in multiple races. Resulting player classification identifies suspicious player behaviour and provides insight into potential improvements to game tuning and cheat prevention measures.

Categories and Subject Descriptors

K.8 [**General**]: Games; I.6 [**Simulation and Modelling**]: Model Development

General Terms

User modelling, Statistical Analysis, Outlier Detection

Keywords

Multiplayer online game, Game telemetry, Analytics, Anomaly classification, Cheating detection

1. INTRODUCTION

Online games have been increasing in popularity over the last several years, including a rise in mulitplayer games. Fairness is especially important in multiplayer games, in order to ensure that all players have the same experience, and prevent players from gaining an unfair advantage through cheating or other means, such as exploiting loopholes. In order to improve fairness, game teams need to understand player behaviour and combat cheating. This is achieved through tuning game play based on analysis of player activity.

Understanding typical player behavior is essential to the process of highlighting areas that need to be tuned. The typical approach is based on statistical analysis techniques to build models of normal behavior and detect outliers. In theory, these outliers can highlight areas of the game that require tuning. However, building accurate models is challenging in practice, as the gathered telemetry data is often contaminated with noise caused by network latencies, server performance issues, or bugs. This erroneous data can skew data models to a point that normal statistical measures such as standard deviation are not representative of the majority of the data. With this skewed data, it becomes difficult to determine not only what is normal, but which outliers are legitimate anomalies in game play.

In this paper, we examine techniques for understanding player behavior using race performance data from an online racing game. Using these techniques, we focus on identifying normal behavior and outliers in this data, and further classifying the outliers using multiple metrics. Outliers of every class might correspond to both cheating and data errors due to issues in data collection or game tuning. With this understanding of outliers, we further show how this information can be tuned to discovering unexpected or unintended behaviors. These behaviors can belong to both players and systems. Additionally, this technique is useful in determining which of the players with outlying behavior are likely to be cheating.

Much research has been done into the problem of defining and detecting cheating in games. In our work, it was helpful to review previous work that defined different measures of cheating[6]. It was also informative to review known cheating methods[4]. Also note, that work has already been done in various areas to combat known cheating methods[5]. While previous approaches focused on how players cheat and ways to combat it, we wanted to diverge our research to focus on how to detect potential cheaters.

In general there are numerous ways to detect outliers[2]. Since the data was mostly normal, the simple IQR method was sufficient. Other work on outlier detection has focused on more advanced mathematical formulas including rank based Bradley-Terry models and statistical limits; however, we chose to attempt a more simplistic approach to see if valid results could still be gained[1].

The rest of this paper is organized as follows. Section 2 describes the data we are working with. Section 3 describes our methods for detecting and classifying outliers in the data. Section 4 shows how the outlier classification can assist in classifying players and detecting suspicious behavior, and section 5 contains our conclusions and some further avenues of research.

2. DATA DISCOVERY

The game data used for this research came from a racing MMO with a few million registered users. A player may have both multiple cars and multiple drivers. Players can customize their cars visually without affecting car performance. As players level up, they unlock higher performance cars for purchase via earned game currency. Players have an option of buying higher performance cars using real world currency regardless of their level in game progression. This game has more than one hundred different tracks that are unlocked with player progression. During game play, drivers select a track to race on and a racing mode, The number of racers in an event is limited to 8. The game supports both player vs player (PVP) and player vs environment (PVE) modes. In PVE mode the player competes against AI racers with similar performance cars. In PVP mode, players are matched to other players based on their experience and car performance. Players can invite their friends in the game to a private race, private PVP or PPVP.

Development team collects large volumes of data representing every aspect of the game. For this research we used a small subset of that data: racing results for players who participated in a challenge event. This data set contains 2.2 million observations for nearly 140 thousand players. Each observation in the data set is a unique instance of a race result. The attributes include player data: the user id, the ids of the car and driver used in the specific race instance, and the driver's level along with the rating score used to match drivers against each other. There is also race data for the instance: the unique id of the instance, the id of the track that was used, and the mode (PVP, PVE, or PPVP). Additionally, there are attributes corresponding to the result information: what place the user came in (1-8, or 0 if they dropped out before the race was completed), the wall time for the beginning of the race as reported by the server, the wall time for the racer's end time as reported by the client, and the computed duration (in seconds) of the racer's time as reported by the client as well as what is seen by the server.

Initial exploration was conducted to build models for the overall data. The distribution of player levels was largely normal with a mean of 20, but a small spike for players at level 10, and a larger spike at level 50, the maximum level. The distribution of player rating scores followed a similar pattern, but with a large spike at the 0 rating. The majority of races were played in the standard PVP mode, with only about a quarter played in PVE or private PVP modes. A histogram for the track number showed no relationship between the id of the track and the number of races, but did highlight several tracks that were played most often, most likely the introductory tracks open to drivers at all levels.

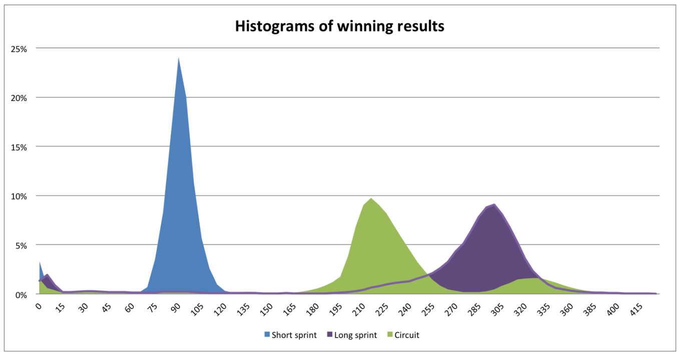

Figure 1: Histograms of winning results

Since it should be expected that each track has a different length, the large data set was grouped by track id for further exploration, and the four most raced tracks were selected for further analysis. It was also determined that the focus could be narrowed to just the results from track winners, both because this is the group where cheating behavior is most likely to present and because the expected range of times for the trailing places can be estimated from the range for first place times.

Analysis showed that the shape of winning time distribution varies widely between tracks and depends on race style (circuit or sprint). Winning times for short sprints are tightly grouped and resemble normal distribution. Distribution of winning times for long sprints is spread widely and is asymmetric. The number of laps for circuit races on the same track varies depending on the use of a "one more lap" power-up. This variability causes winning times to be clustered around two or more peaks.

While in general winning results are grouped in compact 30-50 second intervals, a small number of results is spread across a wide range of up to 1000 seconds. These data outliers correspond to both cheating and data errors.

2.1 Data Modelling

A typical starting point to data modelling is comparing data to a normal distribution. We found that a large number of outliers in the reported times resulted in too large of a standard deviation for the measure to be a useful representation of the data. Our approach is filtering data by removing obvious outliers: results that are unrealistically too fast or too slow. Samples outside of the [0.3* mean, 1.7* mean] range were not used in data modelling. In our experiments filtering removed 4-7% of data records. Results of filtering on mean and deviation are summarized in Tables 1 and 2. Figure 2 demonstrates fitting original and filtered data to a normal distribution. While filtering significantly improves modelling accuracy it fails to capture the overall shape of data distribution.

In order to improve modelling accuracy we investigated segmenting data by race mode and user levels (Figure 3). Comparisons of the winning times of different modes showed that PVP winning times were slightly faster on average than pri-

Table 1: Results From Unfiltered Data

	All Winners	Mean	Std
Circuit	158,022	228.1	84.2
Short Sprint	362,545	102.8	7130.9
Long sprint	100,094	266.9	199.0

Table 2: Results of Filtered Data

	Filtered Records	% of Data	Mean	Std
Circuit	151,636	96%	234.7	45.7
Short Sprint	353,531	98%	92.3	11.0
Long sprint	92,865	93%	283.1	37.6

vate PVP and PVE. This observation supports the idea that multiplayer modes are more competitive than single player. Additionally, there are a significant number of results in both PVP and private PVP modes that are too fast to be considered a legitimate racing time, but not as many for PVE mode. This suggests that competitiveness of multi-player races encourages different behaviours than single player races.

The overall distribution of winning times for long sprints is asymmetric and does not fit a normal distribution. It turns out that grouping the data by driver level yields normal distributions (Figure 4). These distributions are shifted in relationship to each other. This shows that winning times for advanced level drivers are faster than for less experienced drivers. Thus, overall distribution should be modelled as a sum of normal distributions for separate driver levels. While this is true for both short and long tracks, modelling showed that experience, skill and availability of faster cars allowed advanced level players to perform significantly better than less experienced drivers on longer tracks.

Modelling of winning results for circuit races requires separation of results depending on the number of raced laps. Results for the same number of laps should be modelled separately using the same approach as for sprint races. The overall distribution is thus a sum of distributions for races with fixed number of laps. We did not model results of circuit races as the available data does not contain the number of laps metric.

Overall, analysis of winning results showed that racing data could be modelled as a mixture of multiple distributions grouped by track style, racing mode, and driver level. The spread of results across wide time interval reveals presence of outliers. In order to reduce influence of these outliers on numerical results, we pre-filtered data by removing obvious outliers.

3. DEFINING TYPES OF OUTLIERS

3.1 Detecting Outliers

While removal of obvious outliers could be done using heuristic approach, a more accurate outlier detection mechanism is required for further analysis of data. We've chosen to use a statistical outlier detection technique based on interquartile range IQR. Unlike deviation based outlier detection methods[3] [Pearson Ref], IQR technique is suitable for various types of distributions such as asymmetric or mixtures. IQR

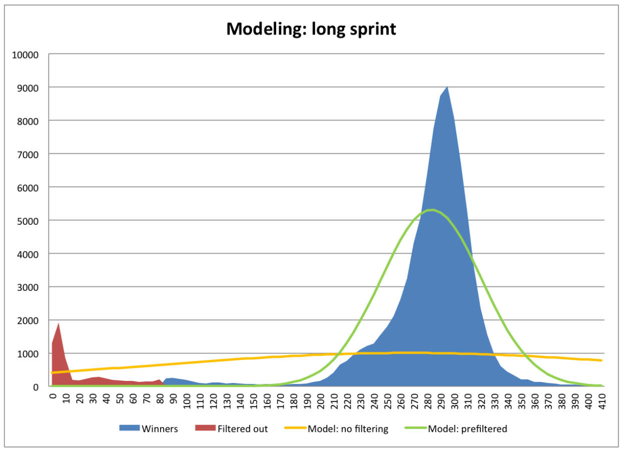

Figure 2: Modeling winning results using normal distribution. Data filtering improves model accuracy.

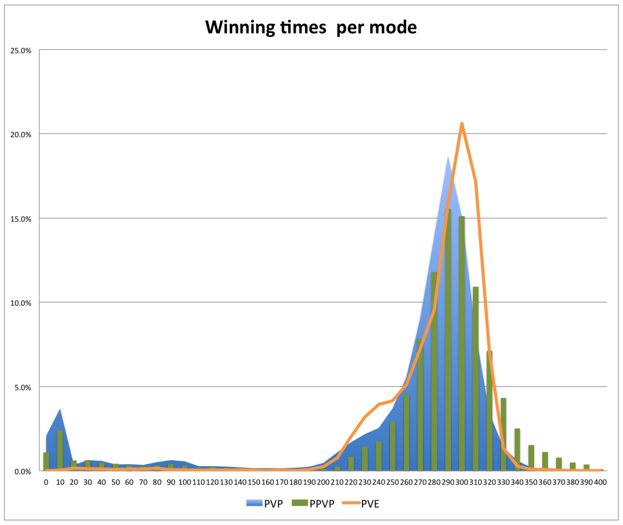

Figure 3: Winning time distributions, aggregated by race mode

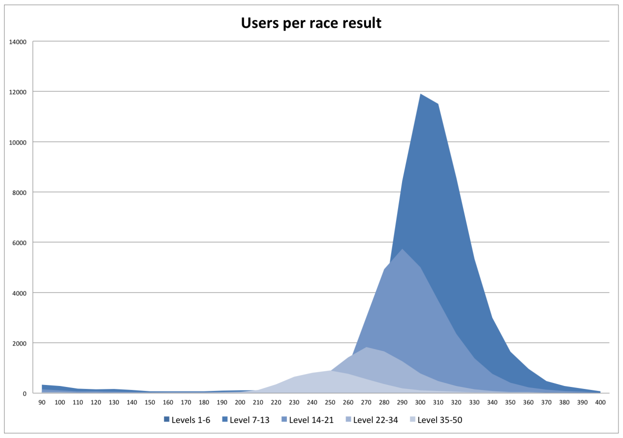

Figure 4: Winning time distributions, broken down by level

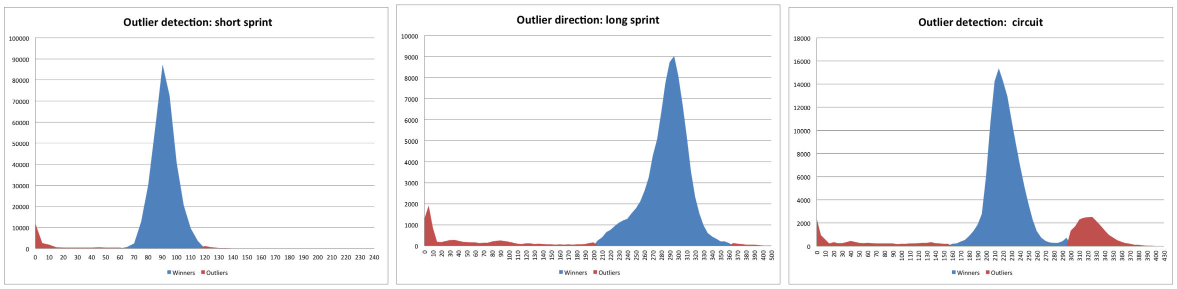

Figure 5: Outliers (in red) found using IQR method

technique is also known to perform well in the presence of data contamination of up to 25%.

Defining x_k as an outlier point, x_U as the upper quartile, x_L as the lower quartile, outliers are any points that satisfy one of the following conditions:

$$x_k > x_U + c(x_U - x_L)$$
$$x_k < x_L - c(x_U - x_L)$$

where c is some constant, typically .5 for symmetrical distributions and 1.5 for asymmetric.

Applying IQR-based outlier detector with $c = 1.5$ to racing data we were able to correctly identify outliers with unreasonably fast times for all types of tracks and races (Figure 5). The detector worked well for both symmetric and asymmetric distributions (short and long sprints). Unfortunately this technique was not reliable in detecting outliers with slow times on circuit tracks. Data corresponding to the additional lap was often mislabeled. This observation demonstrates the need for outlier detection techniques capable of dealing with multiple data clusters. The alternative is treating races with different number of laps independently. Both paths are outside the scope of our investigation. For the rest of the paper we have chosen to analyze data from sprint races only.

The game server collected data on the race duration from the client computer, referred to here as client time, as well as recording the time interval seen by the server itself, or the server time. Data analysis showed that in most records differences between server and client times is less than one second. However, analysis also revealed records with client and server varying by as much as tens of seconds.

After computing the ratios of server times to client times, the data was aggregated by player to obtain a count of how many times an individual user had the same ratio. Ratios that were between 1 and 0.97 were ignored, as these results can all be considered normal ratios, after taking into account latency in collecting the data. Over one hundred players had more than two hundred races with the same ratio between the times, and up to nearly five thousand instances in one case. This strongly suggests that the causes for these differences are consistent across the players. However, with just this information, it is impossible to determine if the reasons for these patterns were due to cheating behavior on the part of the player or data collection errors on the part of the system. In order to make these distinctions, it is necessary to

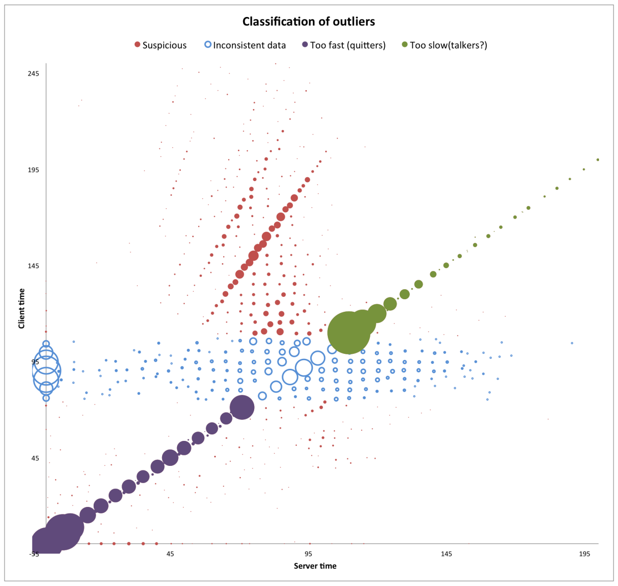

Figure 6: Classification of Outliers

group types of outliers to see how these types are spread among the outlier results of individual players.

3.2 Classification of outliers by client and server time.

The scatter plot in figure 6 summarizes different classification of outliers found using both client and server time metrics. The area of bubbles in the plot represents the number of records with corresponding client and server times. Non-outlier data is omitted for clarity.

We are following the $t$hought that outliers offer an opportunity of discovering useful information related to systems and player behaviours.

We've classified outliers using both client and server time heuristics. The summary of possible causes for generating outlier data were discussed with game engineers and are summarized as follows:

- Too slow: Client time is above the acceptable range and server time is within 1 second from client time. A number of unintended player behaviors could result in slow winning times: e.g. players chatting instead of racing, opponents who quit and leave only one player to finish the race, etc.

- Too fast: Client time is below the acceptable range and server time is within 1 second of the client time. Possible causes:
  - Reporting bugs: race results for players quitting races before finishing the race in some cases were reported incorrectly.
  - Cheating: e.g. teleporting a car to the finish line.
- Inconsistent: Client time is within the acceptable range while server time differs by more than one second. Possible causes:
  - Latency or missing data: delay in processing in delivery or processing of either race start or race end events by the server.
  - Cheating: reported client time is augmented by hacking tools.
- Suspicious: Both client and server time are outside of acceptable ranges. Possible causes:
  - A coincidence of player behaviour (quitting a race or taking too long) and system latency. While both events are relatively rare, they could occur at the same time.
  - Cheating: Known cheating tools allowed the client to speed up or slow down physics simulation by some ratio, which can vary during the race or be fixed, depending on the tool used. If ratio is fixed for the entire race client and server time appear as a multiple of each other. This could explain obvious clustering of some results along a line with slopes 1.2, 1.5, 2, etc.

Outlier detection and classification provides us with additional insights into player behaviour in different race modes. It turns out that wins in single player races (PVE) have the least outliers 3% compared to multi-player modes (PVP=7% and PPVP=12%). Most of PVE outliers are either of "Too slow" or "Inconsistent data" types. This might support initial observation that single player races are least competitive and most of its outliers are results of deficiencies in data gathering. "Too slow" is the largest class of outliers private multi-player races (PPVP). This fact is consistent with the developer's observation that many players were using private races as private chat rooms. Outliers in PVP races are dominated by "Too fast" class and are likely indicative of many players quiting before the finish line avoiding losing the race. It is remarkable that 17% of PVP outliers are of the "suspicious" class compared to 3% and 4% for other modes. This might be indicative of a higher number of cheating in the most competitive multi-player races.

4. PLAYER CLASSIFICATION

4.1 Winners with Unusual Results

With a better idea of how to define normal race data and find outliers, attention was turned to the players. In particular, we were interested in finding a relationship between wins and outliers across players. The premise of this analysis is that outliers caused by system deficiencies are uniformly distributed between all users. While a high number of the outlier race results suggest intentional behaviour of a particular player. Analysis and interview with developers showed that every class of outlier results could be produced by either cheating or deficiencies in data collection. Presence of outliers does not reliably differentiate between winners and cheaters. Instead, outlier detection and classification enabled us to develop player classification metric, **o**utlier ratio: ratio of outliers to all winning results for a player.

We labelled all winning results using previously discussed outlier detection and classification techniques. Further, we computed the number of overall wins of specific class per player. Players with sufficiently high number of wins (20 or above) are classified depending on the ratio of outliers in their winning results. These results are summarized in table 3.

Our analysis reveals that outlier mixture varies across players. In particular, outliers with large miss-matches of server and client time (suspicious class) increase with the overall outlier ratio. In fact, outliers of this class dominate winning results for a small number of players high outlier ratio. In contrast, players with low outlier ratio have nearly even mixture of the other outlier classes and a low number of suspicious cases.

Table 3: Players classified by their outliers, at most every 4th result is an outlier

Winner's Class	Outlier Ratio	% of players
Clean	No outliers	64%
Low Outliers	$< 25\%$	33.5%
High Outliers	$> 25\%$	2.5%

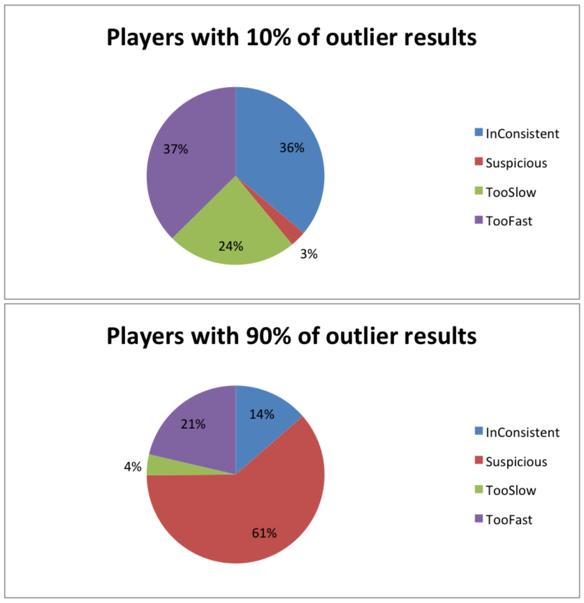

Figure 8: Outlier distribution of players with low or high occurences of anomalies

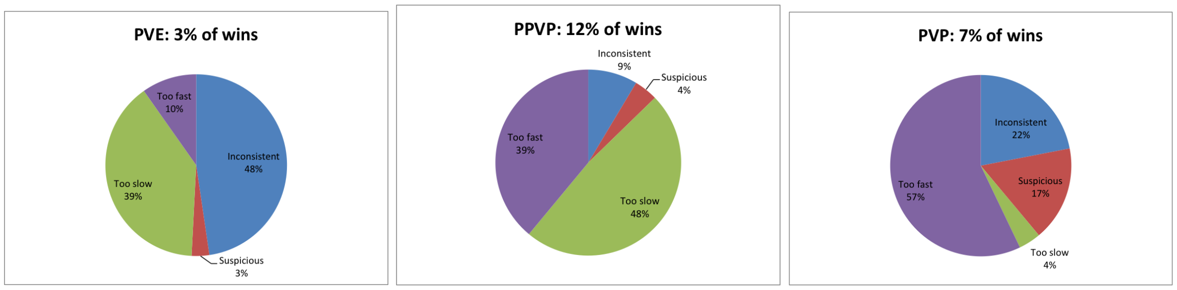

Figure 7: Outliers separated by race mode

Both observations support our hypothesis that data collection problems affect all players equally. High number of outliers in player results indicates player behaviour that might amount to cheating. The especially high number of outliers for a small group of players (2.5%) is particularly suspicious.

4.2 Player career and win outliers

Our analysis thus far was focused on winning results only. We've modelled typical wins and classified outliers. We've shown that outliers are distributed non-uniformly across players, which suggests specific player behaviours in various types of races. Our next step is investigating wins in the context of the player career.

We classified players by the ratio of wins to all races they played. Distribution of players per win ratios for PVP races is presented in Figure 9. For most players win ratio is under 50%. In other words most players win at most one in two PVP races they enter. Only 1.9% of players won more than 70% of races. Such high win ratio might indicate either exceptional skill, better car, intensive use of powerups or cheating.

Figure 10 shows that the number of players with only clean wins (no outliers) decreases with growth of the win ratio. The group of players that have more than a 70% win ratio has the highest number of players with high number of outlier wins. Note that the same group has the highest ratio of outliers to wins: 23%. This is significantly higher than players with lower win ratios, 10%-12%. The mixture of outliers provides additional insights. Outliers of the "too fast" and "suspicious" class peak for the high win ratio group.

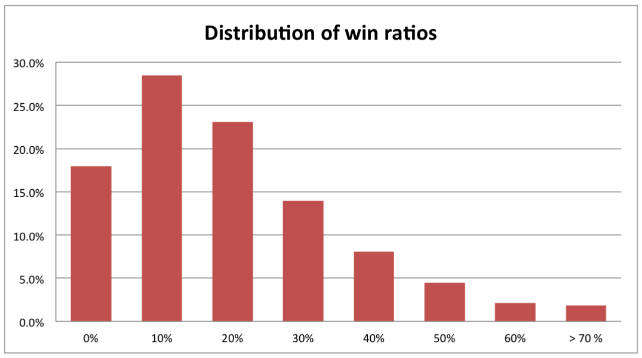

Figure 9: Player Win Ratios

Overall, the analysis above shows that there is a relationship between player win ratios and outlier wins. High win ratios correspond with both the increase of players with outlier wins and high number of outliers per win. This observation suggests that the group of players with high win ratio might contain the higher number of likely cheaters than other groups.

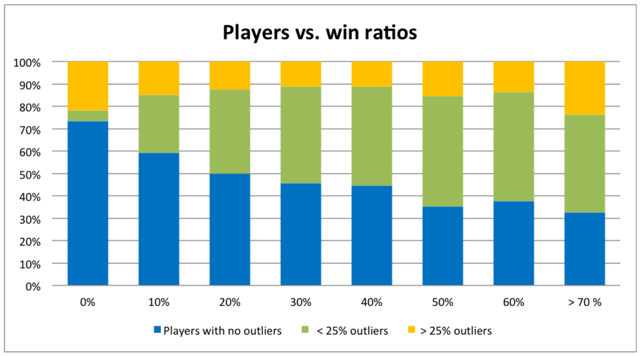

Figure 10: Number of players with outliers increases with win ratio

5. CONCLUSIONS AND FUTURE WORK

Using a combination of visualization and analytical techniques, we have been able to discover several interesting patterns in the game data, which can be explained by several different causes and categorized by the patterns. Furthermore, we have proposed that classifying outliers into groups can be helpful in finding useful outliers, rather than noise produced by data collection issues. These outlier classes can then be used to further the detection of players with suspicious or otherwise anomalous behavior. In our data, these techniques have shown promising results in accomplishing these goals.

Future work on this data includes working to improve the statistical methods used so that non-normal distributed data can be included in outlier detection. Future work in outlier detection in general includes the creation of automated tools to detect and flag suspicious player behavior either on future data sets or in real time.

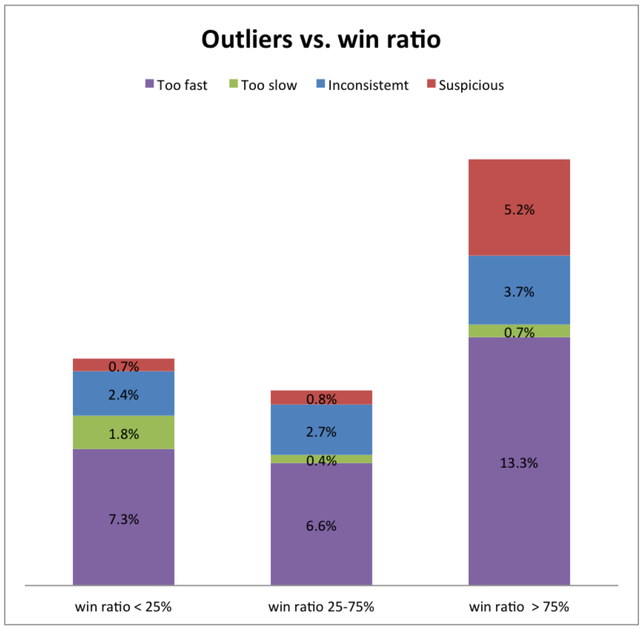

Figure 11: Players with high win ratios tend towards high numbers of outliers

6. REFERENCES

[1] L. Chapel, D. Botvich, and D. Malone. Probabilistic approaches to cheating detection in online games. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 195 –201, aug. 2010.

[2] S. Chawla and P. Sun. Outlier detection: Principles, techniques and applications. `http://sydney.edu.au/engineering/it/~comp5318/lectures/pakddslides.pdf`.

[3] R. Pearson. *Mining Imperfect Data: Dealing with Contamination and Incomplete Records*. SIAM, 2005.

[4] A. Portnoy and A. Rizvi-Santiago. Walking on water: A cheating case study. *Security Privacy, IEEE*, 7(3):20 –22, may-june 2009.

[5] M. Pritchard. How to hurt hackers: The scoop on internet cheating and how you can combat it. `http://www.gamasutra.com/view/feature/3149/how_to_hurt_the_hackers_the_scoop_.php`, jul. 2000.

[6] J. Yan and B. Randell. An investigation of cheating in online games. *Security Privacy, IEEE*, 7(3):37 –44, may-june 2009.