Paper PO-13

# Analytical Approach for Bot Cheating Detection in a Massive Multiplayer Online Racing Game

Andrea Villanes, North Carolina State University

## ABSTRACT

The videogame industry is a growing business in the world, with an annual growth rate that exceeded 16.7% for the period 2005 through 2008. Moreover, revenues from online games will account for more than 38% of total video game software revenues by 2013. Due to this, online games are vulnerable to illicit player activity that results in cheating. Cheating in online games could damage the reputation of the game when honest players realize that their peers are cheating, resulting in the loss of trust from honest players, and ultimately reducing revenue for the game producers. Analysis of game data is fundamental for understanding player behaviors and combating cheating in online games. In this work, we propose a data analysis methodology to detect cheating in massively multiplayer online (MMO) racing games. More specifically, our work focuses on bot detection. A bot controls a player automatically and is characterized by repetitive behavior. Players in a MMO racing game can use bots to play during the races using artificial intelligence favoring their odds to win, and automate the process of starting a new race upon finishing the last one. This results in a high number of races played with race duration showing low mean and low standard deviation, and time in between races showing consistent low median value. A study case is built on upon data from a MMO racing game, and our results indicate that our methodology successfully characterize suspicious player behavior.

## INTRODUCTION

The videogame industry generated nearly $25 billion in revenue worldwide in 2011 [1]. According to the Entertainment Software Association (ESA), the annual sales in the U.S. were $10.5 billion in 2009. Additionally, a strong growth in the video game software market has been seen during the past years, with an annual growth rate by the US game software industry that exceeded 16.7% for the period 2005 through 2008 [2]. Strategy Analytics, a leader in game reporting, reports that online games revenues will account for more than 38% of total video game software revenues by 2013 [3].

Massive multiplayer online games (MMOGs) continue to be a popular sector within the videogame industry. MMOGs support human players competing against others in a virtual world. These virtual worlds are persistent and material worlds, having in most cases a virtual currency that allows players to, for example, buy items that will enhance their gaming abilities [4]. It is in these virtual worlds where cheating exists, but not unchallenged.

Cheating in MMOGs is defined as [5]: “Any behavior that a player uses to gain an advantage over his peer players or achieve a target in an online game is cheating if, according to the game rules or at discretion of the game operator (that is, the game service provider, who is not necessarily the developer of the game), the advantage or the target is one that the player is not supposed to have achieved”.

Cheating in MMOGs is of several types: automation (bots), third-party software that directly affects the game play or modifies the results, and bugs in the game, which creates exploitable loopholes.

Researchers have used machine learning techniques in order to detect cheating and understand players’ behavior in online games. In this work, we study a MMO racing game and analyze race performance data collected over a period of six months to identify suspicious players. A suspicious player is defined as one that might be using additional resources other than his abilities, to win the game. The purpose of this work is to develop a set of guidelines that will serve identifying cheaters in online gaming, who are using bots to gain unfair advantage over other players.

## MATERIALS AND METHODS

This work uses data mining techniques in order to detect suspicious players who might be using bots to gain unfair advantage over their peers. A methodology is proposed in order to analyze data collected from an online racing game, that results in a set of potential cheaters and their characterization. SAS 9.3 and SAS Entreprise Miner 12.1 were used in this paper.

### BACKGROUND INFORMATION

The data for this work comes from a racing multiplayer online game. The game contains around 2.5 million registered users. A user has a unique account identifier. A user can have multiple cars, and multiple drivers. The game consists of a series of tracks. Tracks in the game are unlocked as the player levels up. When the player levels up, they earn

money, which they can use to buy higher performance cars. Players can also use real world currency to buy higher performance cars regardless of their level in the game. The player selects a track to race and a racing mode. A player can have multiple sessions, and multiple races within each session. A session is defined to start at the time when the player signs in their account, and ends when the player logouts of their account. Three racing modes are supported by the game:

- Player vs. player (PVP): players play against other. The number of players in a race is limited to seven.
- Player vs. environment (PVE): the player plays against artificial intelligent players (environment).
- Private player vs. player (PPVP): players invite their friends to play in a private race.

Each mode presents different characteristics that may influence some of the performance metrics (i.e. time in between races). For example, the PVE mode is more deterministic than any of the other two modes because the player is competing against the environment. In this paper, we analyze the races that were played in PVE racing mode.

Time in between sessions would not be an indication of a player using bots, but a consistent time in between races would be. Being a winner is not, but wining a disproportionate number of races is. Participating frequently in the races is not, but participating in a larger than expected number of races may also be symptomatic of suspicious players. A player suspected of being a cheater may show certain characteristics. It is our goal to use performance metrics to uncover patterns of behavior that should help in identifying suspicious players.

## DATA DESCRIPTION

The dataset in this work contained a total of 2.2 million observations for 45,457 distinct players. Each observation in the dataset is a race instance. The dataset contained the following variables as shown in Table 1.

| Name | Range | Description |
|---|---|---|
| LocalUserID | - | Unique identifier of a user |
| EventID | - | Unique ID of a race |
| ModeID | {1, 2, 3} | ID of race mode: PVP, PVE, Private PVP |
| EventSessionID | - | ID of a particular race session |
| ResultValue | {1, 2, 3, 4, 5, 6, 7} | Placing in the race 1-7. 1 is a win |
| EventStartDtm | 20JUL10:00:03:46 - 02FEB11:00:33:14 | Race start time |
| EventEndDtm | 20JUL10:00:05:56.910 - 02FEB11:00:34:35.750 | Race end time |
| EventDuration | 0 - 4294960 seconds | Race duration |

**Table 1. Description of the most relevant data logged in the game**

## PERFORMANCE METRICS

Time in between races: the time it takes a player to start another race upon finishing up his previous race.

Race duration: the time it takes a player to finish a race. Race durations vary depending on the track the player is playing.

## DATA ANALYSIS

The objective of the analysis is to discover suspicious players who are cheating in the game using performance data collected by online gaming companies. Players who might be using bots are of particular interest in this work. We propose the following steps to identify potentially suspicious players: the time in between races, race duration and number of races played are used to differentiate between humans and bots. This can be done because of the consistent repetitive movement that characterizes bots. The following statistics are calculated: (a) median of time in between races, (b) number of total races played by player, (c) number of races played by track and player, (d) mean of race duration by track and player, and (e) standard deviation of race duration by track and player. A “bot player” will present the following characteristics: (a) low median of the time in between races, (b) high number of total races played, (c) high number of races played by track (d) low mean of race duration by track, and (e) consistent race durations by track. Figure 1 presents succinctly the proposed steps to uncover potential suspicious players followed by their description.

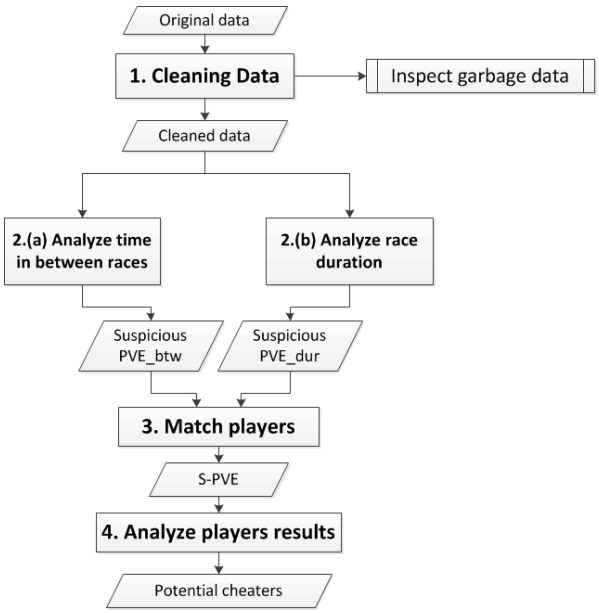


**Figure 1 – Data analysis algorithm**

### 1. Cleaning the data

The purpose of this step is to eliminate noisy data not useful for the analysis. Mean and standard deviation of race duration by player are calculated over all tracks and races. Data points close to the origin (0,0) are of particular interest because these points represent “unusually fast” players with low standard deviation and low mean. These players are either obvious cheaters with all races ending within a few seconds or garbage data. This data is inspected separately before eliminating it from the analysis in order to make sure that the data is not of value for our purposes.

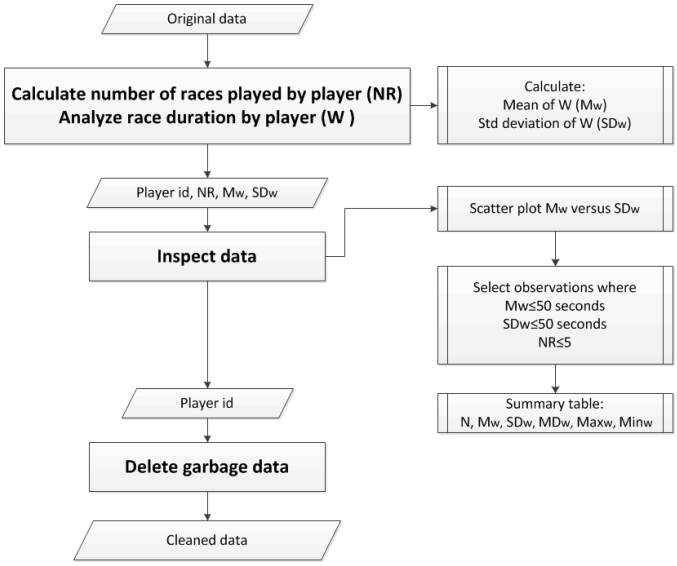


**Figure 2 – Data cleaning algorithm**

In order to avoid eliminating potential cheaters, the number of races played is calculated. Players with a low number of races (i.e. less than five races) are not suspicious of being cheaters, but a player with a high number of races would be, and should not be eliminated as garbage data.

### 2. Analyze the time in between races

The purpose of this step is to uncover suspicious players who might be using bots in order to automate the time in between races. These would be players who had low time in between races, and high number of races played. Two cutoff values: k for the median of the time in between races, and v for the number of races played are chosen in this step in order to identify suspicious players. Suspicious players are the ones who had a median of the time in between races less than k, and a number of races played greater than v.

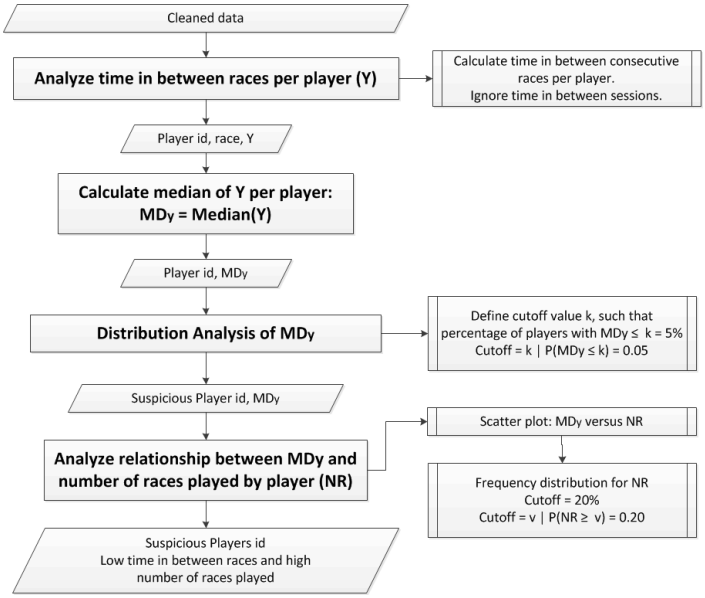


**Figure 3 – Time in between races algorithm**

### 3. Analyze the race duration

The purpose of this step is to uncover players who might be using bots during the races. These would be players who completed the races in a short time, and show the same times consistently across the races in the same track. Three cutoff values: j for the number of races played by player and track, p for the mean race duration and q for the standard deviation of race duration are selected in this step in order to identify suspicious players in this step. Suspicious players are the ones who had a mean race duration lower than p, a standard deviation for race duration lower than q, and a number of races played by track greater than j.

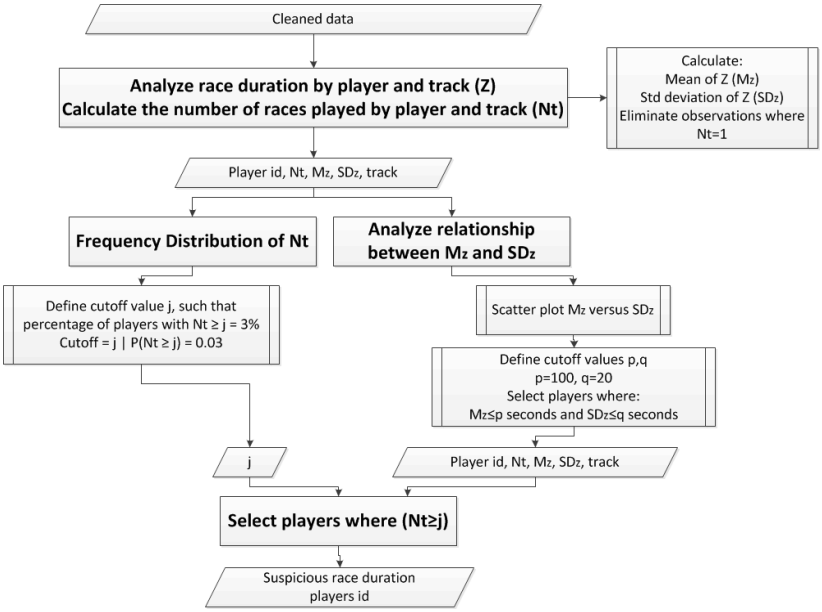


**Figure 4 – Race duration algorithm**

**4. Match players**

The two datasets from steps 2 and 3 are matched by the player id to find out players present in both datasets.

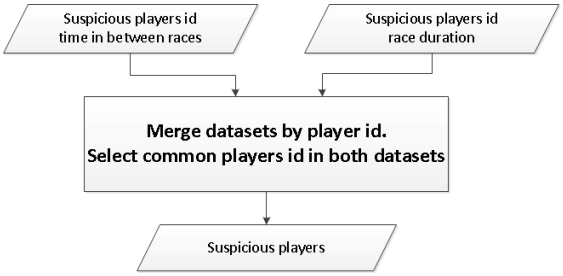


**Figure 5 – Match players algorithm**

**5. Analyze players' results**

The purpose of this step is to characterize potential cheaters, and analyze their placing behavior in the game through summary statistics and expert defined activity rules.

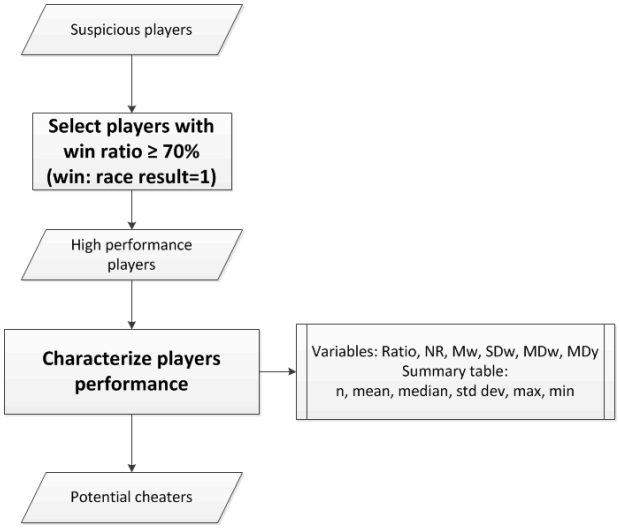


**Figure 6 – Analyze players' results algorithm**

## RESULTS

### Cleaning the data

The number of races played by a player is calculated, and a scatter plot of the mean versus the standard deviation of race duration by player (W) is plotted as shown in Figure 7.

After examination, it was decided that noisy observations would correspond to races with a mean of race duration of less than 50 seconds, and a standard deviation of less than 50 seconds. These are players with smaller times and standard deviations indicating possibly a small number of races, and abandoning the race. Data points colored in red are players that have: (a) Mean of race duration < 50 seconds, (b) Standard deviation of race duration < 50 seconds.

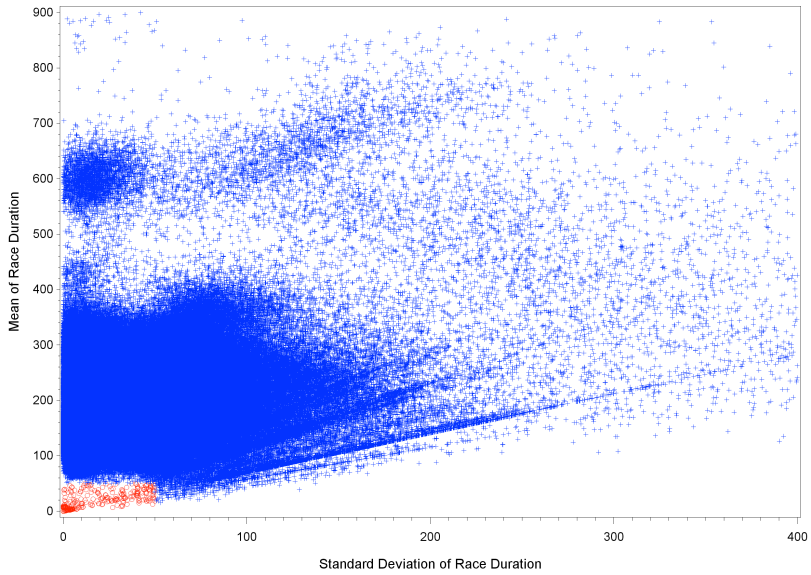


**Figure 1 – Scatter plot Mw versus SDw**

The number of races played by the players in the red cluster is inspected, concluding that 98.68% of the players had five or fewer races played in total. These players are eliminated from the dataset and excluded in the analysis. A total of 4478 players (0.1%) were excluded from the analysis.

**Analyze the time in between races**

The median (MDy) of the time in between races (Y) is calculated per player, and its distribution is analyzed. The cutoff value k for suspicious players is defined, such that:

$$\text{Percentage of players with } MDy \leq k = 5\%$$

$$\text{Cutoff} = k \mid P(MDy \leq k) = 0.05$$

A scatter plot of the MDy versus the number of races played by the player (NR) for those players with MDy ≤ k is analyzed. A second cutoff value v, for the number of races played by each player (NR), is defined for suspicious players, such that:

$$\text{Percentage of players with } NR \geq v = 20\%$$

$$\text{Cutoff} = v \mid P(NR \geq v) = 0.20$$

Players with a low median time in between races (MDy) and high number of races (NR) are selected as a suspicious group based on the time in between races.

Figure 8 shows the distribution of MDy, and the scatter plot of NR vs. MDy. The red dotted line in the distribution of MDy indicates the cutoff value k=40, and the red dotted line in the scatter plot indicates the cutoff value for v=1000. A total of 284 suspicious players based on time in between races are selected.

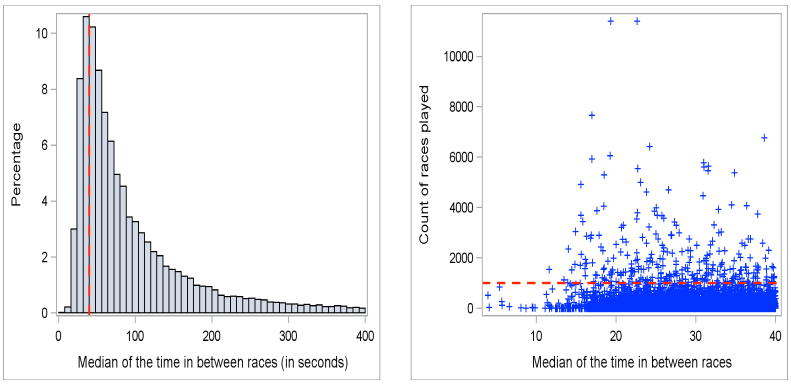


**Figure 8 –Median of time in between races analysis**

**Analyze race duration**

The number of races played by player and track (Nt), the mean of the race duration by player and track (Mz), and the standard deviation of the race duration by player and track (SDz) are calculated. Observations where Nt=1 are eliminated because these observations have SDz=0. A cutoff value j for Nt is defined, such that:

$$\text{Percentage of players with } Nt \geq j = 3\%$$

$$\text{Cutoff} = j \mid P(Nt \geq j) = 0.03$$

A scatter plot of Mz versus SDz is analyzed. Players with the following characteristics are selected as suspicious: a) $Nt \geq j$, b) $Mz \leq 100$ seconds, and c) $SDz \leq 20$ seconds.

Players with a mean of race duration per track (Mz) lower or equal than 100 seconds, standard deviation of race duration per track (SDz) lower or equal than 20 seconds, and a number of races per track (Nt) greater or equal than j, depending on the mode, are selected as a suspicious group based on race duration. The cutoff values are shown in Table 3.

***PVE***

Figure 9 shows the distribution of Nt, and the scatter plot of Mz vs. SDz. The red dotted line in the distribution of Nt indicates the cutoff value j=50, and the red cluster close to the origin indicates those players with a $Mz \leq 100$ and a $SD \leq 20$. Within the red cluster, the players who had a Nt>50 are selected. A total of 1515 suspicious players based on race duration are selected.

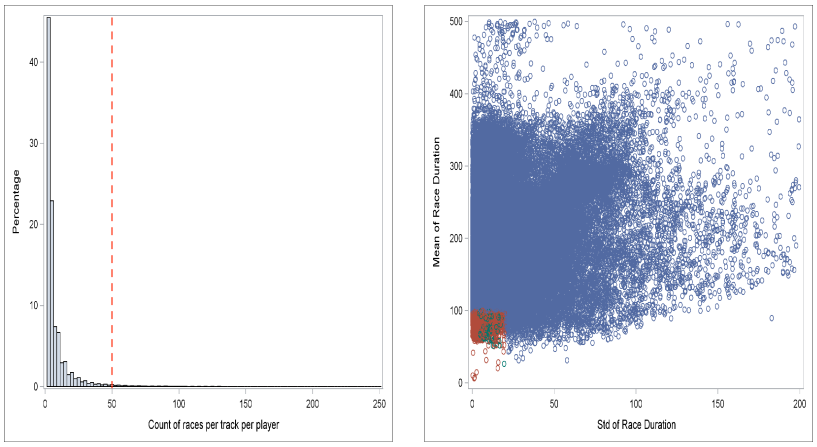


**Figure 9 – PVE: Race duration analysis**

**Match players**

The two datasets from the previous steps are matched by the player id to find out suspicious players present in both datasets. These would be players who had low time in between races, high number of races played, players who

completed the races in a short time, and showed the same times consistently across the races in the same track. A dataset with a total of 214 PVE suspicious players is obtained after merging the two datasets.

**Analyze players' results**

Players with a winning ratio greater than 70% are selected in each of the datasets obtained in step 4. A winning ratio is calculated as following:

Winning ratio= (Number of races where placing result=1)/(Total number of races)

The dataset is then quantitatively characterized in order to describe the potential cheaters. Figure 10 presents the resulting dataset after selecting those players with a winning ratio greater than 70%.

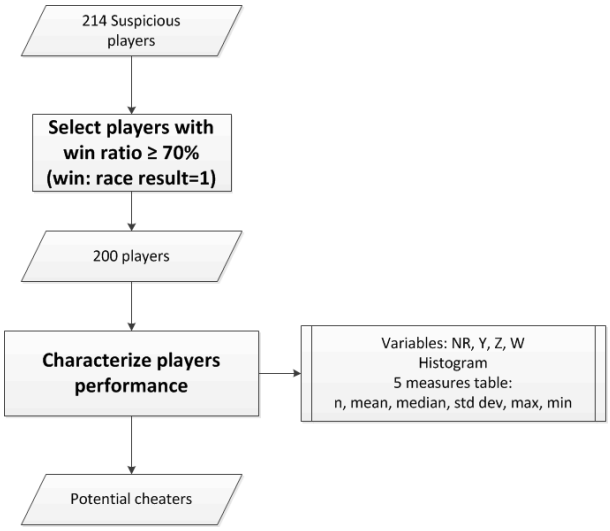


**Figure 10 – Selecting players with win ratio greater than 70%**

In total, 200 players had a winning ratio greater than 70%. A decision tree is used to characterize the data of the 200 players, and to understand their behavior within their group. In order to build the decision tree, a variable "Win" is created, and defined as follows for each player:

- Win = 1: winning ratio > 95%
- Win = 2: 90% < winning ratio ≤ 95%
- Win = 3: 80% < winning ratio ≤ 90%
- Win = 4: 70% < winning ratio ≤ 80%

The dataset was partitioned into two subsets: 70% for training, and 30% for validation. The Win variable is defined as the target variable, and the variables winning ratio (Ratio), the standard deviation of race duration (SDw), the mean of race duration (Mw), the total number of races (NR), and the median of the time in between races (MDy) are defined as the input variables for each of the 200 players. Figure 11 shows the resulting tree.

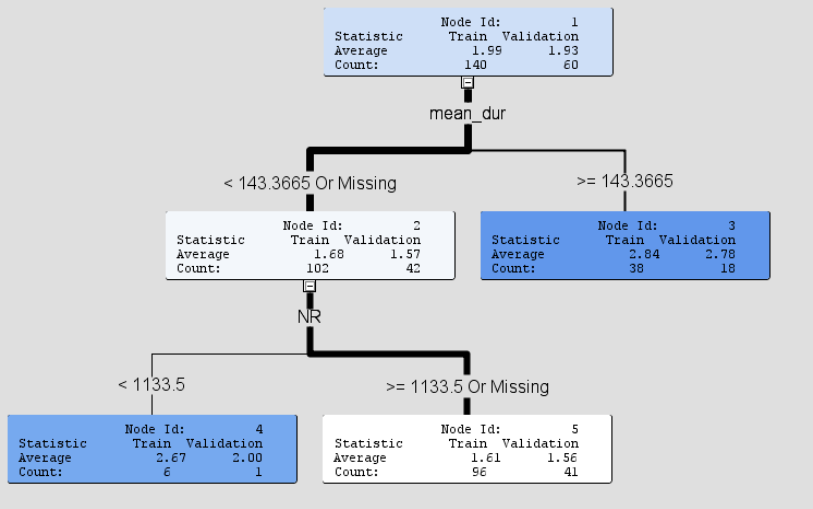


**Figure 11 – Potential cheaters decision tree**

Each of the five nodes is divided into Training and Validation. The 'Average' label shows the average of the target variable 'Win' in the training and validation dataset within the players of that node. The 'Count' label shows the number of players included in that node. Nodes 3, 4 and 5 represent the leaves of the tree. Each path to one of the leaves represent a decision based on the variables defined as the input variables. In this tree, only the variables mean of race duration (Mw), and the total number of races (NR) are significant in order to differentiate between the behavior of the 200 players. In the leaves, the lower the average, the more "suspicious" is the node, since that would mean that the average of the players within that node had a higher winning ratio comparing to the other nodes. In this tree, node 5 had an average of 1.61 in the training dataset, and an average of 1.56 in the validation dataset for a total of 137 players.

## CONCLUSION

In this work, we have presented a methodology to detect bot cheaters in a MMO racing game. Bots perform certain repetitive or precise tasks in place of human gamers. Our methodology builds rules that allow us to identify suspicious players. These rules were based on empirical values after examination of graphs and summary statistics along with expected cheater behavior. A player is considered suspicious of using bots in the game if they present a high number of races played with race duration showing low mean and low standard deviation, and time in between races showing consistent low median value.

In our case study, evidence suggests that bot cheating is present largely in PVE mode of racing, where automation is most effective because of deterministic behavior. Examination of the summary statistics tables for the suspicious players in PVE mode, revealed 200 PVE players with a winning ratio greater than 70%. Furthermore, in PVE mode, the median of time in between races is 26.65, and the mean wining ratio is 0.91. The standard deviation of the median time in between races is 6.84 sec. for potential PVE cheaters and 6.87 sec. for node 5 of the decision tree for PVE potential cheaters, which may be evidence of an automated start after a race ends, which might be an indication of the use of bots in racing. A consistent repetitive movement characterizes bots, and constant standard deviation could be an indicator of the use of bots.

A decision tree has been presented in this work for the PVE mode based on the metrics of the players with a winning ratio greater than 70%. The decision tree provides rules to characterize data into groups. In the decision tree, only the significant variables are used to build the decision rules. In the PVE tree, the mean of race duration (Mw), and the total number of races played (NR) are significant in order to differentiate between the behavior of the 200 players. Inspection of the top five PVE potential cheaters revealed that the winning ratio for these players is between 99.78% (NR=1400) and 99.94% (NR=1737). This is equivalent to only 4 or less races in which the player did not score first place in the races.

## REFERENCES


- [1] ESA Entertainment Software Association. "The Entertainment Software Association - Sales & Genre Data." Internet: http://www.theesa.com/facts/salesandgenre.asp, [Nov. 06, 2012].
- [2] S. Siwek. Videogames in the 21st Century.

http://www.theesa.com/facts/pdfs/VideoGames21stCentury_2010.pdf.

- [3] Strategy Analytics. “Online Game Revenue Fuels Global Video Game Software Market.” Internet: http://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a0=4862, Feb. 19, 2000 [Nov. 06, 2012].
- [4] C. Steinkuehler. Learning in massively multiplayer online games. In Proceedings of the Sixth International Conference of the Learning Sciences, ICLS '04, pages 521-528, Santa Monica, CA, USA, 2004. ACM.
- [5] J. Yan and B. Randell. An investigation of cheating in online games. Security Privacy, IEEE, 7(3):37–44, may-june 2009.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Andrea Villanes
Enterprise: North Carolina State University
E-mail: andrea_villanes@ncsu.edu