바둑학연구, 제17권 제2호, pp. 89-106
Journal of Go Studies
2023. Vol.17, No.2. pp. 89-106

A statistical analysis of amateur go players to assist AI-cheating detection in online go communities

Théo Barollet
Independent Researcher, France
Colin Le Duc
Independent Researcher, France

Abstract: Since the democratization of powerful AI engines for the game of Go, it is not uncommon to see a drastic level increase of some players that must be explained with the help of AI. This is considered cheating and forbidden by most organizations.

When looking at online beginners and stronger amateur players, we discovered that they can display playing strength below professional level and still confidently win the game, as opposed to professional players. This makes using only AI-likeness metrics not sufficient to detect such players. We propose a method based on the analysis of a player's performance considering point loss distribution over several games, taking into account only relevant moves of a game. We still use an AI-likeness metric for analyzing individual games where the use of AI may not be consistent.

We evaluated our methods on two European go official online leagues, where cheating detection was already performed (for a total of about 150 unique regular players, with levels ranging from 20 kyu to 5 European dan). We show that our system confirmed 5 cases of players previously banned for cheating (out of 6). Our methods do not set out to categorize players between “cheaters” and “not cheaters,” but rather rank them in order of suspicion, for the sake of assisting referees and providing them a way to effectively investigate suspicious players over time.

Keywords: baduk, cheating detection, statistical method, online go

I. Introduction

Despite the ancient history of Go and its current prevalence nowadays, especially in Asian countries such as Korea, Japan, or China, it is only relatively recently that it reached other parts of the globe. For instance, the main cultural and technological events that brought attention to this game in Europe are the Japanese manga Hikaru No Go published in 1998, and DeepMind's work on AlphaGo in 2016 [5]. However, most countries and national federations lack a sufficient physical implementation through their territories, while still pursuing the goal of having national rated leagues and, eventually, professional players. In order to allow as many players as possible to play Go in an official way, it is not uncommon for federations to experiment with the creation of online leagues. However, since the recent improvements of AI in the field, cheating at the game, even at a high amateur level, is accessible to most players easily, thus artificially augmenting their rating. This led to leagues and communities to be wary of the integration of online games in official national ladders, either by creating a separate ladder [6] or even by ignoring such games altogether from the official ratings.

Therefore, such federations and affiliated online communities have been creating ethical and fair-play committees, whose goal is to make sure all the games are played in a regular fashion. Unfortunately, more often than not, the number of people doing this work and their available resources are quite low compared to the amount of games that need to be analyzed. This longing for resources and time optimization led to the development of automated tools and methods.

As members of such a committee, our team has been working towards the adaptation and development of such tools, and this paper presents the current

state of our research in using analytical methods to provide useful metrics and information in detecting AI-assisted cheating in amateur games.

Ⅱ. State of the art in anti-cheating detection and related works

Contrary to what exists today in other disciplines such as chess and their FIDE/ACP Anti-Cheating Committee, there is no global organism overseeing cheating detection in Go. Indeed, each server, federation, and online leagues have their own cheating detection mechanisms and there is no global effort to mutualize resources and knowledge.

In fact, due to the lack of resources in some smaller organizations, some leagues and communities do not have any kind of anti-cheating systems at all, making them vulnerable to cheating, and sometimes preventing them from offering online rated games to their players (American Go Association, IGLO).

1. Related works

To our knowledge, only a few articles focus mainly on cheating detection in Go.

One of them from Egri-Nagy and Törmänen [1] tries to detect AI-assisted play with a single SGF file. We share some common hypothesis with their work:

Cheating detection cannot be made in a fully automatic way without getting many false positives, a human intervention is needed [3].

Their cheating-detection method is also based on several metrics derived from AI go engine and the combination of several suspicious metrics make them conclude an AI is used.

However, they quickly tackle the problem of looking at several games for a single player, by suggesting to detect a sudden increase in player strength in a single SGF-file to detect cheating. We believe this cannot detect many cheaters and that it would instead require a long term analysis. Our method uses many games of a player's history because we are not always able to detect such sudden increases. For example, a newer strong player in a league may already be a strong amateur player or a cheater. We also encountered the case of a player cheating for a long time and mimicking a plausible increase of player's strength over time.

Most of the cheaters we detected in the context of our cheating-detection work would have the benefit of the doubt of being strong players if we could only look at a single record of their games.

The other article from Park et al. [2] provides a way to compute an AI-likeness metric. Obvious moves are filtered out from the game, as well as moves played when the game is "decided'' (more than 95% win rate for either player). The remaining moves are considered "meaningful" and are considered "AI-like" when the score difference between the top AI-move and the played move is below a certain threshold. They apply their method to professional games and manage to observe a significant difference between top professional players and known cheaters in terms of "AI-likeness".

We adapted some of their methods in our work, for example by filtering moves and by computing an AI-likeness metric for a whole game. However, we cannot directly apply their method for several reasons. Firstly, they use a closed source AI engine and their score metric is derived from some internal

values of their AI engine that are not standard in publicly available engines. Secondly, their metric is especially suited for professional players where a cheater would need to play very closely to AI-level in order to beat top professional players. In our amateur level context, a cheater can play many sub-optimal moves and still confidently win the game.

Ⅲ. Dataset and methodology

As members of an online club affiliated to the french go federation, our internal league made for a good practice ground to test and evaluate our methods, as the games played are rated on the national ladder. This online league, that has existed for 3 years now, gathers around fifty players monthly, each of whom play 3 games in that time period. That accounts for 1776 games at the time of writing.

The anti-cheating detection committee has detected 6 players with strong confidence over the past 3 years. This is the result of long term analysis of players' games with moderation tools used by go servers to detect cheating.

Once the committee believes the player strength cannot be explained without the help of an AI, the player is contacted and a meeting is planned. Only 2 players admitted cheating (at that time, we hired an European professional player to analyze the games; and he found that the performance displayed would be of a player above his own professional level, thus leading to the conclusion that an AI was used) and the other ones did not provide convincing explanations of their strength and refused to play over the board games. Only after these meetings took place and their refusalof playing over the board games (even friendly games) were they accused of cheating and

suspended from the league. These cases serve as a reference baseline against which we can compare our findings and are identified in relevant figures by the “flagged” hue (the orange points).

For each player taking part in the league, we gathered up to 100 of their online games outside of the league, with the following filters :

- Ranked on the server ladder
- No handicap games
- No correspondence games

This brings our games count to 7225. Each of these games is then analyzed with Katago [4], an open-source go engine. Most of these analyses have been performed by AI Sensei [8], which is an online platform allowing players to execute free analysis up to 50 visits per move. Therefore, this is the number of playouts that we opted for in our own analyses, as this is the most likely settings that could have been used for cheating during live rated games, and because such a setting is still enough to beat all of the players included in our dataset, with levels ranging from 20 kyu to 5 dan on OGS.

An interesting side effect that occurred during the making of this dataset is that some games played against robot players ended up appearing. Such artificial players, many of which have been artificially made weaker to be of acceptable challenge against amateur players [7], are detected as suspicious players by our models without any intervention on our part, thus supporting our findings.

Ⅳ. Statistical analysis of amateur online games

In this section, we look at two different metrics and see if we can discrim-

inate known cheaters from our dataset, as well as organizing other players according to these metrics.

1. AI-likeness metric

We adapted the method described in the Park et al. paper to be used with a different engine and in amateur games. The main difference is that we do not have access to the same metrics as their engine is proprietary. However, using the score lead AI estimation in Katago proved to be pertinent as a relative metric between moves. While we envisioned to use the utility metric provided by the analysis engine, which is derived from both the winrate and score lead metrics, the author has confirmed to us that this is not a pertinent metric to compare different moves, as there is no relation between turns with this metric.

In adapting the original paper to be implemented with this metric, we calculated that the threshold for considering a move to be "AI-like" is 0.6 on the score lead metric. Indeed, despite seeming high in the context of professional games, most amateur games present wide ranges of point loss within their moves, and choosing enough moves within the 0.6 score lead variation can still confidently lead to a win.

Another difference with the Park et al. paper is that we do not discard early game moves in our analyses, as the amateur players present in our dataset do not possess such a strong knowledge of the early games sequences as professional players. The repartition of moves considered to be "AI-like" after the various filters described in the Park et al. paper is shown in Figure 1. By plotting the AILR metrics with the winrate of each player included in our dataset, we are able to confirm that most of the known cheating players are

gathered in the top-right corner of the plot, as shown in Figure 2.

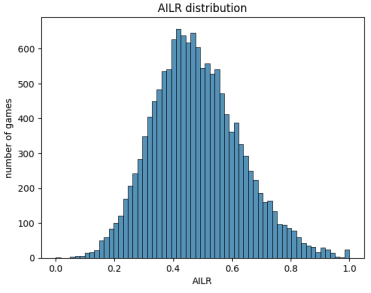

Figure 1

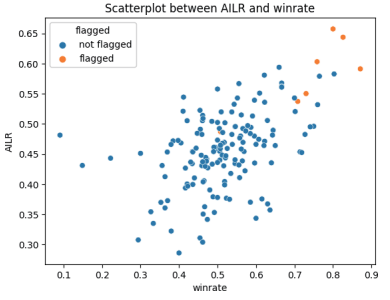

Figure 2

2. Move error metric

We detected several cheaters that do not blindly follow AI top moves. They often do not play the best moves but they almost never make mistakes, especially for amateur players and throughout many games.

We will look at the amount of points lost after each move compared to the top AI move and its distribution over several games. We consider the logarithm of this quantity, because the difference between a mistake of 1 and 2 points and 14 and 15 points is not the same in terms of impact on the game outcome.

Examples of distributions for this metric can be seen on Figure 3. These are only illustrative examples but the trend we can see in those cases is constated in the entire dataset: the stronger the player, the moreit looks like a decreasing exponential with a higher steepness. The convex parts that can be observed for the 10k player can also be observed for players around that strength as well as players with a lower level.

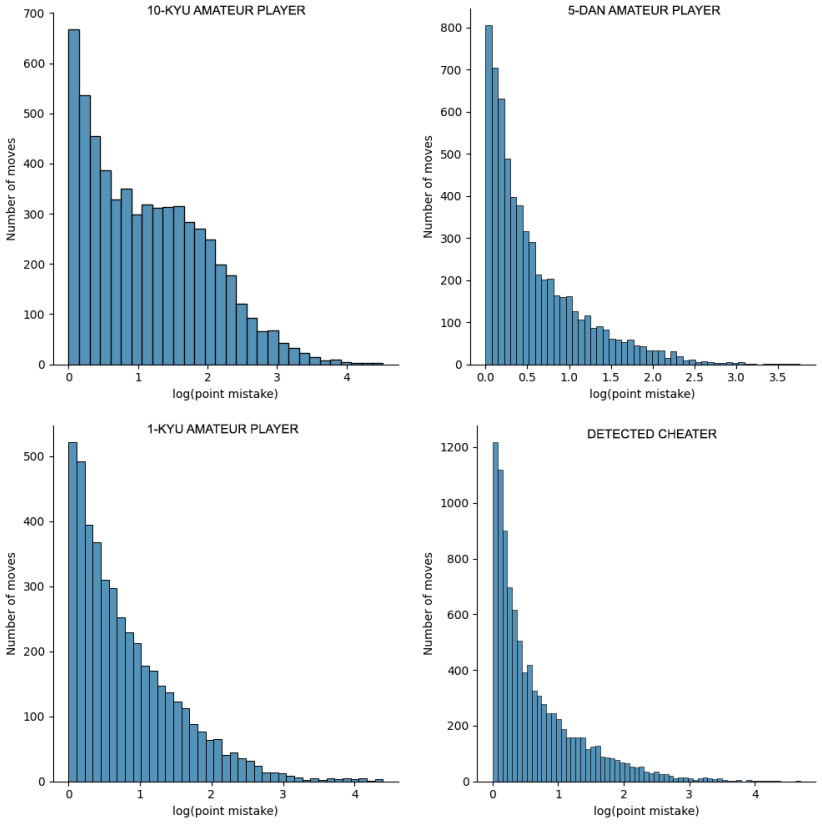

Figure 3: Examples of distributions of the logarithm of the point mistake for 4 different players.

First, we will look at the simple average for each player over all of their moves. This can be seen on Figure 4. The cheater in the cluster of supposingly non-cheating players is a player who got a sudden increase in strength and suddenly have beaten several dan players while being around 6k for a long time. We see that apart from this player, the cheaters we already detected all have one of the fewest mistakes among all players. Two players in this sus-

picious cluster are considered non-cheating, as they are already known dan players who have been playing over the board tournaments for a long time. The rightmost, bottommost point is a player with only 5 games in our dataset. Even if they could be qualified as suspicious, we would not consider this to be sufficient to qualify the player as "suspect" unless some other metrics are also suspicious.

The main issue with this method is that a player with a few games and supposingly not cheating can display values greater than known cheaters, and there is no clear and definite boundary between suspicious and non-suspicious games. We expend on this method to determine a more deciding criterion.

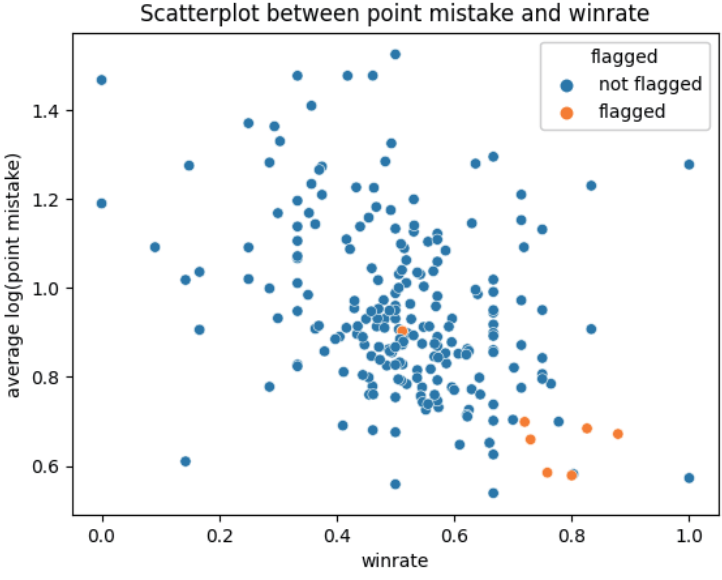

Figure 4

Our goal is to approximate the distributions seen on Figure 3 as an exponential curve. We define two coefficients a and b such that a*exp(-b*x) fits the distribution. A high value of b coefficient means that the exponential decreases quickly, and that the player only makes a few mistakes. As for players with lower ranks, where an exponential curve should not fit the distribution, we apply the fitting anyway and observe the variance ofthe parameters that should be especially high.

As an example, the parameters and their variance of this fit for our 4 examples can be seen on Table 1. We see that the value of the b coefficient (how fast the exponential decreases) is correlated with the strength ofa player, at least in our 4 examples. The values for the whole dataset can be seen on Figure 5.

	a	variance (a)	b	variance (b)
10k player	635	1113	0.65	0.00247
1k player	563	53	1.07	0.00039
5d player	840	188	1.87	0.00185
Cheater	1256	720	1.89	0.00324

Table 1: Fitting coefficients and their respective variance to approximate the error distributions as an exponential curve.

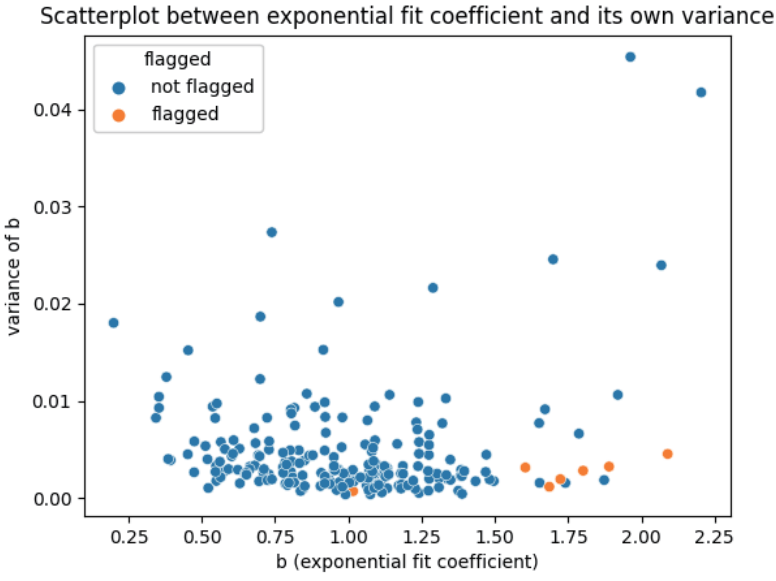

Figure 5

Compared to the previous method, we observe that the boundary between the “flagged” and “not flagged” cluster is clearer, although the two stronger dan players are still in the “flagged” cluster. The aforementioned outlier with only a few games in our dataset is not beyond the “flagged” cluster anymore. However, a supposedly non-cheating player appearedin this cluster, this player is a strong dan player who only plays a few online games. It is too early to become really suspicious about this player but this may suggest the need for further investigation in the future.

V. Discussions

The method described in the previous part has a major weakness that would need to be addressed: it cannot discriminate between a cheater and a strong dan player. This is where conventional and non-analytical methods come into play: for example, we may ask strong dan players in this cluster to play some over the board games ifthey are not already doing so. Moreover, it cannot detect players with sudden strength increases, but this can be detected if we look at each game individually and we see a major difference between some metrics.

This would also prove useful in discriminating against a player who only cheats in a few of their games. By using our metrics on only a few games that are believed to be of particular interest (such as rated league games), we can accidentally bias our results due to the potentially low number of games in that subset.

However, these metrics and the described methods can still be helpful in developing tools to assist fair-play committees, by gathering player-specific analyses efficiently, such as the AILR evolution over time shown in Figure 6.

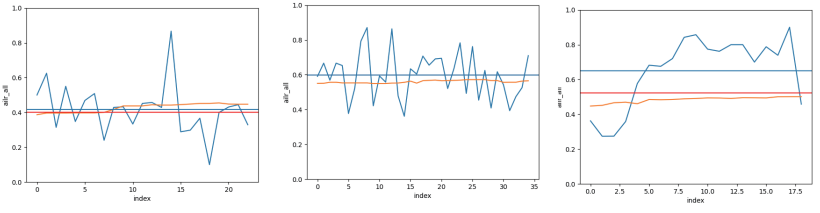

Figure 6

While the information conveyed in this figure is not directly helpful in detecting suspicious players or games, it can still provide useful information for referees when they investigate specific events, saving them some time and energy by automating this data collection.

VI. Conclusion and future works

We can expand on this method by including more games in our dataset, especially leagues with higher level players, and trying to take rating into account to discriminate against suspicious players more efficiently. By using an open-source engine and collaborating with other international leagues and go servers, we hope to offer a greater range of tools to them and allow them to independently improve on this method.

If we manage to gather more information on cheating players and games where cheating occurs, we should also be able to develop new methods that cover a greater range of cases and more subtle cheating, as well as per-player statistics even more useful for fair-play committees' investigations.

The findings in this research reinforced our knowledge of the benefits and limits of using analysis detection for amateur players, as other methods need to be developed as well, especially in the domains of game servers tooling and social investigation processes.

By releasing this paper and the associated code publicly, we hope our work can inspire other organizations to adopt a similar process with medium or long-term analysis to avoid false accusations as much as possible, and, once enough elements are unfortunately gathered, allow them to quickly contact

alleged cheaters to confirm the suspicions, encouraging them to play over-the-board games or to meet with other players.

Ⅶ. Acknowledgments

A huge thanks to AI Sensei for providing analyses of our dataset, as well as Benjamin Teuber, Robin and Aldo for the inspiring discussions and inputs. Thanks to Olivier for his proofreading of the initial abstract.

We also want to thank the following organisms who helped sponsor this research : the French Go Federation, the Stones In The Shell online go club, and the French Online Go League.

References

[1] Egri-Nagy, A., & Törmänen, A. (2020, November). Derived metrics for the game of Go–intrinsic network strength assessment and cheat-detection. In 2020 Eighth International Symposium on Computing and Networking (CANDAR) (pp. 9-18). IEEE.

[2] Park, J., Im, J., On, S., Lee, S. J., & Lee, J. (2022). A statistical approach for detecting AI-assisted cheating in the game of Go. Journal of the Korean Physical Society, 81(12), 1189-1197.

[3] Barnes, D. J., & Hernandez-Castro, J. (2015). On the limits of engine analysis for cheating detection in chess. Computers & Security, 48, 58-73.

[4] Wu, D. J. (2019). Accelerating self-play learning in go. arXiv preprint arXiv:1902.10565.

[5] D Silver, A Huang, CJ Maddison, A Guez, L Sifre… nature, 2016 Mastering the game of Go with deep neural networks and tree search

[6] Decision to create an "hybrid" ladder to take into account online games in some national rating system : https://ffg.jeudego.org/informations/officiel/cr/CR_CA_20210930.pdf

[7] An OGS robot player designed to play at a level of 1 kyu : https://online-go.com/player/652529

[8] Online go games analyses service AI Sensei: https://ai-sensei.com/

Received: 15, Oct, 2023

Accepted: 30, Nov, 2023