Explainable AI for Cheating Detection and Churn Prediction in Online Games

Jianrong Tao, Yu Xiong, Shiwei Zhao, Runze Wu, Xudong Shen, Tangjie Lyu, Changjie Fan, Zhipeng Hu, Sha Zhao, *Member, IEEE*, and Gang Pan, *Member, IEEE*

***Abstract*—Online gaming is a multibillion dollar industry that entertains a large, global population. Empowering online games with AI has made a great success, however, ignores the explainability of black-box model makes AI less responsible and hinders its further development. In this article, we introduce and discuss the audience and the concept of XAI (eXplainable AI) in online games. We propose a GXAI workflow, which combines the strong expressiveness of multiview data sources and the clear transparency of multiview black-box models. We present four specific classifiers and explainers in the character portrait view, the behavior sequence view, the client image view, and the social graph view. Experiments conducted on real-world datasets for game cheating detection and player churn prediction show the accuracy of classification and the rationality of explanation. We also discover and present numerous interesting and valuable findings from the individual, local, and global explanations. We implement and deploy three practical applications, including evidence and reason generation, model debugging and testing, and model compression and comparison in NetEase Games and have received quite positive reviews from user studies. More future work is in progress since this is the first work that introduces XAI in online games.**

***Index Terms*—Cheating detection, churn prediction, explainable artificial intelligence, industrial application, interpretable model, online game.**

I. Introduction

ONLINE gaming is one of the most successful applications having a large number of players interacting in an online persistent virtual world through the Internet. Over the last decade, empowering online games with AI has become a trend due to the rapid development of AI technology and successful applications in various industries.

AI can help creating a new generation of gaming experience for players in online games. The development of AI technology can also be promoted with the massive data and simulation environment of the game platform. AI in online games has resulted in a paradigm shift along multiple dimensions.

1) *Versatility* enables fitting a wide range of functions in an increasing number of domains in online games like game chatbot, game test, battlefield AI, matchmaking, recommender systems, cheating detection, churn prediction, virtual characters, etc.
2) *Expressiveness* unlocks the data modalities (tabular data like character portraits, sequential data like behavior sequences, image data like client images, graph data like social graphs, etc.) and enables joint/multi-model applications in online games.
3) *Adaptability and Efficiency* through custom optimized hardware like GPUs and tensor processing units (TPUs) enabling AI practitioners to train bigger models faster and cheaper even small data regimes through transfer and multitask learning.

In the quest for more accurate AI in online games, the availability of compute resources and increasing dataset sizes have fueled a trend toward more complex nonlinear models. On the flip side, these more complex models have become increasingly opaque coupled with the fact that they are still fundamentally built around correlation and association, have resulted in several challenges.

1) *Spurious Correlations* can be learned from data, often hampering the model's ability to generalize and leading to poor results.
2) *Loss of Control* due to practitioners' reduced ability to locally adjust model behavior in problematic instances.
3) *Proxy Objectives* resulting in large differences between how models perform offline, often on matching proxy metrics, compared to how they perform when deployed in the game online.
4) *Undesirable Data Amplification* reflecting biases that do not agree with our norms in the game domain.
5) *Loss of Debuggability and Transparency* leading to low trust as well as the inability to improve the models and outcomes. Furthermore, this lack of transparency impedes adoption of these models, especially in online games.

Manuscript received 23 November 2020; revised 30 November 2021 and 11 March 2022; accepted 3 May 2022. Date of publication 10 May 2022; date of current version 16 June 2023. This work was supported by Natural Science Foundation of China under Grant 61925603. *(Jianrong Tao and Yu Xiong contributed equally to this work.) (Corresponding authors: Yu Xiong; Runze Wu.)*

Jianrong Tao and Zhipeng Hu are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, and also with the Fuxi AI Lab, NetEase Inc., Hangzhou 310052, China (e-mail: taojianrong0@zju.edu.cn; zphu@corp.netease.com).

Yu Xiong, Shiwei Zhao, Runze Wu, Xudong Shen, Tangjie Lyu, and Changjie Fan are with the Fuxi AI Lab, NetEase Inc., Hangzhou 310052, China (e-mail: xiongyu1@corp.netease.com; zhaoshiwei@corp.netease.com; wurunze1@corp.netease.com; hzshenxudong@corp.netease.com; hzlvtangjie@corp.netease.com; fanchangjie@corp.netease.com).

Sha Zhao and Gang Pan are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: szhao@zju.edu.cn; gpan@zju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TG.2022.3173399.

Digital Object Identifier 10.1109/TG.2022.3173399

2475-1502 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

The above challenges almost always occur when AI is applied to various important fields in online game. For example, in game cheating detection, to avoid killing normal players, game operators are cautious in using the predictions of AI models, although the predictions are highly accurate. In game churn prediction, the game operators push the players with high churn risk. Due to the lack of specific churn reasons, the effect of push is unsatisfactory [1]. The field of XAI has seen great potentiality in online games, since we desire the ability to explain black-box models in understandable terms to a human. With the ability to understand, explain, and control models, the usefulness and fairness of AI in online games will be gated. Our previous work, XAI-driven explainable game cheating detection [2], has been accepted as the best paper by CoG 2020. This article extends the application of XAI in game churn prediction. The contributions of our study are fourfold.

1) *XAI in Online Games:* To the best of our knowledge, this is the first work that introduces XAI in online games.
2) *The GXAI Workflow:* We introduce specific classifiers and explainers from the portrait/behavior/image/graph view and describe explanations from the individual/local/global aspect.
3) *Real Evaluations:* We evaluate game cheating detection and player churn prediction in online games with real-world datasets. Extensive experiments show the accuracy of classification and the rationality of explanation. Numerous interesting and valuable findings have been discovered and presented.
4) *Real Applications:* Three applications including evidence and reason generation, model debugging and testing, model compression, and comparison have been implemented and deployed in NetEase Games and received quite positive reviews.

II. Related Work

A. AI in Online Games

Online games have been linked with AI since the first program was designed to play chess. The challenge to defeat human expert players, for example, AlphaGo [3], the first to beat the world go champion, DeepStack [4], who beat professional players for the first time in heads-up no-limited Texas Hold'em, and JueWu AI [5], who beat top professional players in *Honor of Kings*, has greatly advanced the AI research, affecting breakthroughs in, e.g., computational intelligence, machine learning, reinforcement learning, and combinatorial game theory. In turn, such new AI methods have been used in online games, for instance to game matchmaking [6], to automatic game testing [7], to game character autocreation [8], to automatic skin binding [9], to recognize and remove toxic behavior from the player chat channels [10], or to realize virtual person generation [11].

B. XAI in Industry

There were many state of the art XAI applied to critical systems in industry. Google Cloud[1] released Explainable AI Beta, which had a set of tools and frameworks to help AI practitioners develop interpretable and inclusive machine learning models and deploy them with confidence. Thales XAI Platform[2] was designed to provide explanation for a ML task (classification, regression, object detection, segmentation) and provided different levels of explanation, e.g., example-, features- based, counterfactual using textual, and visual representations. ZETANE[3] accelerated concept to production with a visual, interactive work space for AI, which helps debugging, explaining, and optimizing AI models.

III. Problem Definition

A. Game Cheating Detection

The game industry has suffered serious threats from game cheating, and game cheating detection remains one of the most urgent problems that game publishers need to address [12]. Game bots in massively multiplayer online role-playing games (MMORPGs) [13] are automated programs that reach the system kernel and perform continuously tough or tedious tasks without requiring the rest periods. Accordingly, game bots can easily achieve great superiority over honest players, leading to the huge imbalance in the in-game ecosystem. The perspective plug-in is common in first-person shooter games (FPSs) [14], which adds several plug-in elements to interface game. Players can get huge information invisible in the game and take advantage of this to win the game. It causes unfair competition and harms the honest players.

B. Player Churn Prediction

We defined players who have logged in games in the past three days but will not log in for the next seven days as seven-day churn players. With the saturation of the online game industry, the cost of player acquisition is getting higher and higher. Player churn has became one of the biggest problems in the gaming industry [15]. In consequence, how to accurately predict player churn and analyze the corresponding reason is quite important. We could develop targeted interventions based on the churn reasons to take actions against players, who are about to churn. For example, we could get the flaws in the game design and update the game version iteratively. We could also give props packages to the players who are about to churn that meet his needs and interests. This is effective to retain players and increase the gameplay stickiness.

C. XAI in Online Games

As shown in Fig. 1, XAI is targeted at model developers like experts/scientists/engineers responsible for designing and delivering AI models. The goal is to leverage AI explanations to ensure/improve product efficiency by simplifying model development as well as debug models by explaining the model's behavior. Domain knowledge can be discovered by understanding the model itself. Product managers, business leaders, and end

[1][Online]. Available: https://cloud.google.com/explainable-ai

[2][Online]. Available: https://www.thalesgroup.com/en

[3][Online]. Available: https://https://zetane.com/

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

Fig. 1. Diagram showing the different purposes of explainability in AI models sought by different audience profiles in online games. Two goals occur to prevail across them: need for model understanding, and regulatory compliance.

users may also find XAI useful in online games. Game players affected by model decisions may understand their situation and verify fair decision from the AI explanations. As mentioned above, it is far from enough that AI models reach high prediction accuracy for game cheating detection or player churn prediction. Game publishers and operators pay more attention on the evidence and reason generation. Could we give evidence that this player is cheating or normal? Could we give reasons for this player churn or retention? To answer these questions, XAI in online games seems to be the only solution.

IV. Dataset Description

A. NetEase Games

Living up to the company's motto of "good games have no borders," NetEase Games[4] has made great popularity and traction in domestic and overseas markets. In 2018, revenues of NetEase's online gaming segment reached $5.845 billion, up 10.8% year over year. NetEase Games continues to maintain high growth rate and deliver new hit games. To commit to the pursuit of the highest quality games and player experience, NetEase Games has developed and published dozens of popular games especially MMORPGs on PC and mobile, including Ghost II Online, Ghost Mobile, Fantasy Westward Journey Mobile, Fantasy Westward Journey Online, etc. Other online games produced by NetEase Games are also famous worldwide such *as Onmyoji, Identity V, Rules of Survival*, etc. We collected more than 300 billion different types of game logs from *Justice Online*,[5] *Meteor Heroes*,[6] and *Knives Out Plus*[7] which contain more than 40 million character creation activities. *Justice Online* is one of the most successful MMORPGs in China, which created a breathing virtual world. *Meteor Heroes* is a famous ACT, which inherited the hard-core spirit of action games and raised the response requirements to the extreme. *Knives Out Plus* is a next-generation FPS, which restored the natural and built environment and incorporated more Chinese elements in design.

B. Logs in Online Games

Various activities performed by the characters or state change events of the clients are recorded in the form of structured game logs through the servers. The logs consist of the following information:

1) *Timestamp:* when a specific activity or event occurs.
2) *Log ID:* an ID that identifies the type of character activity (hunting monster, trading virtual money, client screenshot, etc.) or the state change event type (level up, virtual money increasing, mouse trajectory moving, etc.).
3) *Source character's information:* the character's role ID, gender, level, class, virtual money, knead face time, etc.
4) *Target character's information:* the target character's information if the character interacts with another character.
5) *Detailed information:* depending on each activity or event type, detailed information related to the activity or event.

We collected the datasets from *Justice Online, Meteor Heroes* and *Knives Out Plus,* which are popular games released by NetEase Games. The datasets are now available for downloading.[8] Considering the privacy, the characters are ensured by anonymizing all personal identifiable information.

C. Character Portraits Construction

We built a universal game character portrait system from real-time game logs, which covers multiple different topics, including account information, basis information, social information, task information, gameplay information, item information, money information, etc. We have extracted 523-d and 181-d portraits from the character portrait system of *Justice Online* and *Meteor Heroes* respectively. Supplementary Tables 1 and 2 show 30 important character portraits.

D. Behavior Sequences Construction

Each character's behavior sequence is composed of lists of events ordered by time stamp, which contain following three features:

1) *Event ID:* the event conducted by a character, for example, using a certain skill, obtaining a certain item, etc.
2) *Timestamp:* when the event of the character occurs.
3) *Level:* the current game level for the character. The lowest level of each player is 1 and the highest is raised regularly.

Fig. 2 visualizes the behavior sequence of a cheating player in *Justice Online,* which gives us a general idea of how behavior sequence looks alike. Each slot represents a event, and different events are assigned different colors. There are a total of 85 751 different events in *Justice Online,* and 10 617 different events in *Meteor Heroes.* The average length of player behavior sequences

[4][Online]. Available: http://game.163.com/
[5][Online]. Available: https://n.163.com/
[6][Online]. Available: https://lx.163.com/
[7][Online]. Available: https://hyp.163.com/
[8][Online]. Available: https://github.com/fuxiAIlab/GXAI/tree/master/data

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

Fig. 2. Behavior sequence example of cheating player who frequently takes game tasks (orange color) and picks up items (red color).

(a) Game client image examples (b) Transaction in Justice Online

Fig. 3. Client image and social graph examples.

in *Justice Online* is 13 220 while 5459 in *Meteor Heroes.* Supplementary Tables 3 and 4 show 30 important character behaviors.

E. Client Images Construction

For the game cheating detection task in *Knives Out Plus,* we intercepted the game client's picture at the moment the player shoots and kills the opponent, so that we could determine whether the player has used a perspective plug-in based on whether there are nongame elements on the screenshot. For the player churn prediction task in *Justice Online,* we recorded the player's mouse movement trajectory every 0.4 s in the game log. We have integrated the mouse's track point into the player's client screenshot. By analyzing the client's mouse trajectory, we could determine whether the player will be lost from the game next. To protect the privacy of players, we only take screenshots of the resources in the game. Two game client image examples are shown in Fig. 3(a).

F. Social Graphs Construction

We construct four different types of graphs from the game logs of *Justice Online* and *Meteor Heroes,* which include a transaction graph, a friendship graph, a teaming graph, and a chatting graph. A graph example is visualized in Fig. 3(b).

The transaction graph shows assets exchange relations between characters in the virtual world. Edges indicate the virtual currency of established transactions between characters.

The Friendship Graph is built upon unidirectional friendship in online games. A character can send an invitation to another character and remove friends from his friendship lists.

The Teaming Graph is made up of collaborative relations between characters. A team is temporally formed with the same goal and is disbanded after achieving the goal.

Fig. 4. Proposed GXAI Workflow.

TABLE I
CLASSIFIERS AND EXPLAINERS

View	Method
Classification	
Portrait	RF, XGBoost, LightGBM, CatBoost [17], etc.
Behavior	CNN, LSTM, BLSTM, ABLSTM, Transformer [18], etc.
Image	VGG, GoogleNet, ResNet, DenseNet [19], etc.
Graph	GCN, GAT [20], etc.
Explanation	
Portrait	**TreeSHAP** [16], Anchors [21], etc.
Behavior	**DeepExplainer** [22], NLPExplainer [23], etc.
Image	**GradientExplainer** [24], CAM [25], Grad-CAM [26], etc.
Graph	**GNNExplainer** [27], GraphLIME [28], etc.

The Chatting Graph expresses the communication relationship between characters. A character can send a private message to other characters for individual communications.

V. GXAI WORKFLOW

We present the GXAI (**XAI** in Online **G**ames) workflow in Fig. 4, which contains three core modules: game data construction, classifiers and explainers, XAI applications for audiences. *Game Data Construction*. As shown in Section IV, game data are extracted from game logs and represented in many different views, e.g., tabular (character portraits), sequential (behavior sequences), visual (client images), or graphical (social graphs).

Classifiers and Explainers.[9] The algorithm we employed for classification and explanation might depend on the use-case, model type, data format, etc. We selected the post-hoc explaining for a given classifier that started with a black box model and probed into it with a companion model to create explanations. Table I shows the classifiers and explainers in our GXAI workflow. The explainers attribute a tabular model's prediction to its features, attribute a sequential model's prediction to individual words/behaviors, attribute a visual model's prediction to its pixels and attribute a graphical model's prediction to its subgraphs. What's more, one explanation does not fit all and here we proposed three types of explanations.

[9][Online]. Available: https://github.com/fuxiAIlab/GXAI/tree/master/src

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

1) *Individual Explanation* shows the explanations associated with individual predictions (i.e., what was it about the features of this player that made her leave the game).
2) *Local Explanation* summarizes the local groupness from individual explanations. We generally sampled 2000 samples and performed supervised clustering [16] and found large number of very explanatory groups/clusters.
3) *Global Explanation* shows the entire predictive model to help user understand it (e.g., global feature importance, whether obtained directly or in a post-hoc manner).

XAI Applications for Audiences: The black-box model continues to provide the accurate prediction while explanations help improving human inspections. Explanation needs vary depending on the user type who needs it and also the problem at hand. We demonstrated several novel and practical applications below, which successfully apply XAI in online games.

1) *Evidence and Reason Generation* for the game operator teams and game designer teams by exposing attributions. They observed our generated individual explanation and judged whether it is sufficient to form an evidence for game cheating or form a causal reason for player churn. When a player makes a banned appeal, the game operators can show the model explanations as evidences of cheating. Game designers can make phone calls to investigate the cause of player churn or directly intervene with players to verify whether the reasons are reasonable.
2) *Model Debugging and Testing* are helpful to many different audiences. Game AIer teams could apply the model's explanations to discover spurious correlations and label leaks and make timely adjustments to the model's data or structure. As for mispredictions, we could attribute a model misclassification to the features, which are responsible for it. We brought model testing in the model development process. For one thing, game testing teams could use their norms and principles in the game domain to discover undesirable data amplification. For another, the explanations of historical model versions could be compared to find out whether the differences between the models are significant, thereby preventing loss of control of online models.
3) *Model Compression and Comparison:* Feature attribution values are commonly used to identify the most important features used in model prediction. We can employ these to perform model compression from the data and structural level. For example, we use only the most considerable portraits to train the tabular model, and only the most important behaviors or fragments to train the sequential model, and crop the unimportant image pixel areas to train the visual model. By modifying training samples or features, we use less data and smaller model structures to train better models.

VI. Experiments

A. Experimental Settings

We partitioned the datasets and obtained the nonoverlapped training set (80%) and testing set (20%) respectively. More details about the dataset segmentation can be found in Supplementary Tables 5 and 6. All the models were trained on the training set, and we executed the grid search strategy to locate the best parameters on validations by fivefold cross validation. We measured the performances on the testing set with area under the curve (AUC) and ACC (accuracy) for both game cheating detection and player churn prediction. The training time, inference time (for all the testing set), and explaining time (for one sample on average) were also recorded for the computing efficiency comparison. We explained all the models in different views and showed many interesting findings.

B. Performance Comparison

Table II shows the performance comparison of the game cheating detection and player churn prediction among different competing methods. XGBoost achieves the highest AUC and ACC among all the methods in portrait view. Transformer is optimal among all the methods in behavior view. GoogleNet outperforms other methods for game cheating detection and DenseNet exceeds for player churn prediction in the image view. Trade_GCN transcends other methods for game cheating detection and Friend_GAT goes beyond other methods for player churn prediction in the graph view. Table II also demonstrates the significant training time, inference time, and explaining time, which indicates that problems that were previously intractable for exact computation are now acceptable.

C. Explanations for GCD

1) Portrait Explanations:

a) Individual portrait explanations: The cheating player in Fig. 5(a) has a large computer physical memory size, and it is very likely that the cheating studio is opening a good deal of accounts in batches. At the same time, his average percentage of storyline watching is very small since cheating players are not as interested in the storyline as normal players. They are more concerned about how to profit from the game at a low cost that leads to his small proportion average daily PVP duration in the past month. The most obvious evidence for the normal player in Fig. 5(b) is that he has more chats. His average number of chats on the team, the number of emojis used, and that of comments on the world channel within two weeks are higher.

b) Local portrait explanations: Four clusters are found for game cheating detection in Fig. 6(a). For the players in the cheating cluster 1, their average percentage of storyline watching are all 0. For those in normal cluster 2, they use more silver in the game. In another cheating cluster 3, the players have larger computer physical memory size, and the average percentage of storyline watching is relatively low. At last, players in normal cluster 4 conduct more chats.

c) Global portrait explanations: As demonstrated in Fig. 7(a), the cheaters' computer physical memory and display memory are generally large, and are basically high-end equipment of cheating studios. Normal players are more inclined to watch the game storyline, and spend more time pinching their faces. The cheating players rarely chat, while normal players not only speak in the team, but also make comments on the

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

TABLE II
GAME CHEATING DETECTION AND PLAYER CHURN PREDICTION PERFORMANCE COMPARISON AMONG DIFFERENT METHODS

Algorithm	AUC	ACC	Training Time	Inference Time	Explanation	Explaining Time
Portrait View						
RF	0.9764, 0.9192	0.9516, 0.8989	31s, 5s	0.2408s, 0.2486s	Portrait Explainer	0.0591s, 0.0131s
XGBoost	**0.9876**, **0.9254**	**0.9724**, **0.9006**	**9m38s**, **3m40s**	**0.0004s**, **0.0005s**	Portrait Explainer	**0.1745s**, **0.0999s**
LightGBM	0.9859, 0.9243	0.9659, 0.8998	53m10s, 42m39s	1.0286s, 1.3269s	Portrait Explainer	0.0226s, 0.0133s
CatBoost	0.9834, 0.9234	0.9674, 0.8967	23m21s, 9m28s	0.1652s, 0.4948s	Portrait Explainer	0.0034s, 0.0022s
Behavior View						
CNN	0.9667, 0.9131	0.9561, 0.8874	1h32m40s, 2h4m13s	1m20s, 2m5s	Behavior Explainer	23s, 16s
LSTM	0.9389, 0.9120	0.9376, 0.8935	4h10m56s, 5h3m49s	2m42s, 3m29s	Behavior Explainer	33s, 28s
BLSTM	0.9506, 0.9048	0.9418, 0.8884	7h30m27s, 10h6m49s	3m24s, 4m55s	Behavior Explainer	1m5s, 54s
ABLSTM	0.9661, 0.9105	0.9507, 0.8881	8h11m52s, 12h46m49s	4m24s, 5m44s	Behavior Explainer	1m54s, 1m36s
Transformer	**0.9836**, **0.9227**	**0.9694**, **0.8998**	**10h53m17s**, **14h19m33s**	**6m9s**, **7m26s**	Behavior Explainer	**2m35s**, **2m18s**
Image View						
AlexNet	0.5259, 0.6534	0.6727, 0.4812	1h36m53s, 1h16m46s	3m27s, 1m50s	Image Explainer	15s, 4s
VGG	0.5030, 0.7430	0.7605, 0.7399	55m43s, 41m55s	1m33s, 1m12s	Image Explainer	2m39s, 2m2s
GoogleNet	**0.9644**, 0.7000	**0.9147**, 0.5329	**1h12m17s**, 53m25s	**5m10s**, 1m50s	Image Explainer	**1m2s**, 24s
ResNet	0.9351, 0.6918	0.8940, 0.4992	42m25s, 33m43s	1m26s, 58s	Image Explainer	3m6s, 2m51s
DenseNet	0.9268, **0.7843**	0.9065, **0.8531**	1h25m40s, **29m38s**	10m19s, **1m48s**	Image Explainer	5m54s, **1m39s**
Graph View						
Trade_GCN	**0.9832**, -	**0.9795**, -	**2h50m3s**, -	**48s**, -	Graph Explainer	**2m53s**, -
Team_GCN	0.9759, 0.8096	0.9715, 0.7960	2h44m22s, 1h59m57s	53s, 54s	Graph Explainer	3m12s, 7m34s
Chat_GCN	0.9742, 0.7816	0.9584, 0.7868	2h58m39s, 1h37m17s	49s, 55s	Graph Explainer	2m44s, 3m53s
Friend_GCN	0.9808, 0.8139	0.9713, 0.8085	2h48m11s, 1h23m5s	48s, 58s	Graph Explainer	3m37s, 4m24s
Trade_GAT	0.9669, -	0.9537, -	6h3m36s, -	45s, -	Graph Explainer	3m37s, -
Team_GAT	0.9576, 0.8045	0.9516, 0.7956	6h57m47s, 3h32m24s	42s, 37s	Graph Explainer	4m2s, 8m48s
Chat_GAT	0.9581, 0.7386	0.9495, 0.7891	6h5m47s, 2h33m59s	46s, 34s	Graph Explainer	3m47s, 5m8s
Friend_GAT	0.9598, **0.8554**	0.9525, **0.8171**	5h59m6s, **2h38m44s**	46s, **34s**	Graph Explainer	3m55s, **5m59s**

Fig. 5. Individual explanations with SHAP values which indicate the attribution of portraits and behaviors. Colors in plots are smoothed as the larger the SHAP value, the redder the color, and the smaller the SHAP value, the bluer the color. (a) Cheating with individual portrait explanation. (b) Normal with individual portrait explanation. (c) Churn with individual portrait explanation. (d) Retain with individual portrait explanation. (e) Cheating individual behavior. (f) Normal individual behavior. (g) Churn individual behavior. (h) Retain individual behavior.

Fig. 6. Supervised clustering with individual explanations identifies among 2000 individuals distinct subgroups of players that share similar reasons for specific roles. Each prediction was explained using individual explanations, and then clustered using hierarchical agglomerative clustering (imagine a dendrogram above the plot joining the individuals). Red feature attributions push the score higher, while blue feature attributions push the score lower (as in Fig. 5 but rotated 90'). (a) Game cheating detection with local portrait explanation. (b) Player churn prediction with local portrait explanation. (c) Game cheating detection with local behavior explanation. (d) Player churn prediction with local behavior explanation.

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

Fig. 7. Global explanations of XGBoost models and Transformer models on game cheating detection and player churn prediction. Every individual is run through the model and a dot is created for each feature attribution value, so one player gets one dot on each feature's line. Dot's are colored by the feature's value for that player and pile up vertically to show density. (a) Game cheating detection. (b) Player churn prediction. (c) Game cheating detection. (d) Player churn prediction. (e) Game cheating detection. (f) Player churn prediction.

world channel, and send some emojis. The cheaters regularly create teams and take part in more side missions. Since cheating players do not have strong operability like normal players, the percentage of mission failures and abandoned missions is relatively high. Normal players prefer to involve in PVP gameplay such as SJTX. They are also more active in guild activities. For players with large guild funds, the normal possibility is relatively large, but there are also guild groups with concentrated cheating players. Equipment upgrading is one of the core gameplay of the game. Although the proportion of cheaters participating in this type of gameplay is relatively large, normal players improve their equipment even faster.

2) Behavior Explanations:

a) Individual behavior explanations: The cheating player in the red block in Fig. 5(e) continuously accepted, completed, or abandoned the task, and gained money through it. In the dark red part at the end of the behavior sequence, this player began to frequently transfer money to other accounts. The normal player in the blue block in Fig. 5(f) participated in some guild activities, and made comments on many channels. There is a dark red part in the middle that he left the guild, later he joined a new guild soon.

b) Local behavior explanations: The results in Fig. 6(c) indicate that there are three clusters in game cheating detection. The cheating cluster 1 has been punished for remote procedure call (RPC) cheating, accelerator cheating, and GM setting. The normal cluster 2 is characterized by having recharged to buy VIP and fashion clothes, etc. The players in normal cluster 3 participate in PVP gameplay such as BDGSJ, etc.

c) Global behavior explanations: As illustrated in Fig. 7(c) and (e), only normal players will normally trigger

Fig. 8. (a)–(c) Explanations in image view. (a) Game cheating detection. (b) Local and global image explanations.

antiaddiction system reminder, and are willing to talk to other players during the game. The cheaters have had RPC cheating punishment, accelerator cheating punishment, and GM setting punishment. They constantly receive quests to improve experience, and consume physical energy more and faster. The cheating players continue to profit from the game while normal players will spend a moderate amount of money in the game. Generally, only normal players will recharge to buy VIP and buy fashion clothes in the game. Normal players will pay attention to their ranking and take part in some gameplays like BDGSJ. As cheating players use special game bots, there will be some bugs related to scene loading or skill releasing interval, which will be recorded in the game logs too.

3) Image Explanations: Fig. 8(a) shows the important pixels of the model to determine the player cheating. The player uses the perspective plug-in to mark the position of the opponent with a red frame, and at the same time, the player's ID and the remaining health are marked above the frame. We cluster important pixels in the individual explanations, and we observe several typical clusters in Fig. 8(b). First, the plug-in interface

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

Fig. 9. (a)–(f) Explanations in graph view. (a) Game cheating detection with individual transaction explanation. (b) Player churn prediction with friendship explanation.

appears in the upper left corner. Second, the plug-in interface appears in the lower right corner. Third, it exists a large number of red or green frames indicating the position of the opponent just above the middle of the plug-in interface. We aggregate the locations of important pixels. On the whole, the game content display panel and minimap above the game interface have no plug-in elements, and neither do the game operation panel and chat window below the interface.

4) Graph Explanations: From the filtered subgraph of Fig. 9(a), we can observe a typical real money trading. The node 14 is a gold banker, who gathers virtual goods from the gold farmers and sell them to gold buyers. The nodes 3,6,11,16 are gold buyers, who purchase virtual money with real money from the gold banker. The node 4 is a gold farmer, who works full time playing by using automated programs or by hiring low-cost laborers. Similarly, in the explanations of friendship and teaming, cheating players are always in the same friend or team circle. In the explanations of chatting, we found the gold buyer constantly consults gold bankers for money trading.

D. Explanations for PCP

1) Portrait Explanations:

a) Individual Portrait Explanations: In the recent cycle, the number of days the churn player in Fig. 5(c) has logged in is few. In the long term, he has logged into the game for a lot of days in a month. Judging from this, he has only recently begun to show a tendency to churn. Analyzing the reasons for his churn, from a level perspective, he has been fully experienced in the game. He has a relatively small width of device and has only a 50% chance of winning the $1v1$ PVP gameplay LJXX. Frustration and failure in the game lead to a poor game experience. The retain player in Fig. 5(d) is a very loyal player who has logged in every day for the past month from the long-short term. He wins all QYH, a PVE gameplay in the game, and really has a high sense of achievement.

b) Local Portrait Explanations: Turning to the local explanations on player churn prediction in Fig. 6(b), it reveals five clusters. All the players in retain cluster 1 are all online in the last three days and the week. As for those in retain cluster 2, they come online more actively in the long-short term. Players in the retain cluster 3 are on top of the leaderboard, since their game win rate is close to 100%. Contrary to those in retaining cluster 2, players in churn cluster 4, the number of online days is relatively small. And finally the churn cluster 5 gathers the players whose online duration of the last day is 0.

c) Global Portrait Explanations: Fig. 7(b) presents that the fewer days a player logs in within a week or a month, the more likely he is about to lose. Moreover, the player logging in for many times within a month has also a great possibility to churn. It is most likely because he has already fully experienced the game content and feels bored. As the number of days a player account is created becomes longer, his character level becomes higher, and the player stays online longer, which means a more loyal player and a higher probability of retention. The players with wider device is more likely to retain, while those who have longer device tend to churn. Since *Meteor Heroes* is a landscape game, a high width-length ratio of the device will help to achieve a better operation experience. It is very interesting to note that a higher win rate of the QYH gameplay conduces to increase the retention probability. Nevertheless, a higher win rate of the $3v3$ PVP gameplay QKT attributes to the high churn probability. We also found that a minority of players have a high win rate in QYH gameplay, but they also tend to churn. Similarly, players who play QYH gameplay more often are more willing to retain, but the more times they play SBZ PVE gameplay, the more likely they are to churn. Counter intuitively, as for the players, who have a large number of friends, it still turns out a high churn probability. The rank of Sword Battle is extremely high, as well as the game recharge amount. Both of them may lead to churn or retention.

2) Behavior Explanations:

a) Individual behavior explanations: As illustrated in Fig. 5(g), the churn player logged into the game. He had both wins and losses in the first few matches, but the red block indicates that he lost several games in a row during this period of time, and then there were multiple matches that have dropping off or network delay issues. In Fig. 5(h), the retain player in the blue block first participated in some daily and guild tasks after logging in. That a red block appears in the middle shows this player joined the battle rank during this period, but the ranking continued to decline.

b) Local behavior explanations: Four clusters in Fig. 6(d) emerged from the player churn prediction experiment. The churn cluster 1 refers to those, who have the problems such as disconnection, reconnection, and network delays. The retaining cluster 2 is featured by having purchased various items in the shop. The churn cluster 3 has encountered the game failures and gradual decline in points or rankings. The players in retain cluster 4 actively take part in daily, dispatch, and guild tasks.

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

TABLE III
USER STUDIES FOR EVIDENCE AND REASON GENERATION (POSITIVE RATE)

Job	PortraitExplainer	BehaviorExplainer	ImageExplainer	GraphExplainer
Game cheating detection				
Game operator teams	95.9%	90.1%	96.4%	81.1%
Game designer teams	83.1%	76%	88.6%	68.3%
Player churn prediction				
Game operator teams	90.3%	84.8%	60.4%	94.2%
Game designer teams	73.7%	73.1%	53.8%	89.5%

c) Global behavior explanations: In Fig. 7(d) and (f), active players log in and out more frequently and are more stickier to the game. Players who have purchased items in the game shop are also more likely to stay in the game. Churn players tend to have gone through some bad gaming experiences such as disconnection, reconnection, and network delays. It is interesting to notice that players who participate in daily quests, dispatch quests, and guild quests are more likely to stay, while players who frequently participate in weekly quests are more likely to churn. It is somewhat counter-intuitive that we often quickly lose players who participate in reflux training missions. Even those who open regularly gift packages are easily churn. Players who take part in daily check-ins, receive mission rewards, and actively participate in in-game PVP battles and win are more motivated to play. However, those players whose points or rankings gradually decline will soon churn from the game.

3) Graph Explanations: It is interesting to note the follow-up effect in the game, especially in team formation and chat. Once the players with whom they usually have a frequent team relationship or have a very close chat relationship are churn from the game, they often tend to follow the churn. Fig. 9(b) illustrates that the nodes 80 110 129 have been churn from the game, and the nodes 1106 are about to churn on stop playing the game, so our model judges that the churn probability of node 117 is remarkable. This is a common phenomenon in the game. Once a large number of players in the friend circle begin to lose, the probability that they follow the churn tends to be very high. We cluster the filtered subgraphs in Fig. 9, and it is noticeable that there can also be different structure types under the same relationship. These different types of structural information can be used as a rule to directly search for cheating players or find causal reasons for churn players.

VII. APPLICATION

A. Evidence and Reason Generation

Figs. 5, 8, and 9 in Section VI have shown the individuals' cheating evidences and churn reasons from different views. To validate that the explanations are the most natural and intuitive, we ran user studies of game operators and game designers. Participants were not selected for AI expertise. Evaluation criteria for explanations consists of truth, usefulness, relevance, coherence with prior belief and generalization. Table III presents the feedback of the user studies for evidence and reason generation. We collected about 2000 pieces of feedback, and received about 84.9% of positive feedback for GCD explainers and 77.5% for PCP explainers. The GCD image explainer reached a maximum of 92.5%, while the PCP only reached 57.1%, which means that the GCD image explanations are more reasonable and intuitive, but the PCP image explanations based on player's mouse movement trajectory are not easy understand and need further research. The feedback from the game designer teams is slightly worse than that from the game operator teams, but it also reached 75.8% on average, and the graph explainer reached a maximum of 89.5%. These are quite far more than expected.

(a) Label=Normal, Pred=Cheating (b) Label=Normal, Pred=Cheating

Fig. 10. (a)–(b) Model debugging with the image explainer.

B. Model Debugging and Testing

From Fig. 7(b), we found that most of the portraits that are important for churn prediction are not the causal features, but the correlation features. Although adding these features can more accurately predict player churn, it also increases the difficulty of cause analysis. We deleted the features related to player churn expression. Although the accuracy of the model prediction slightly decreased, we could analyze more valuable real reason related features. As for game cheating detection in Fig. 7(c) and (e), there are some records of punishment for cheating. These behaviors leak the label of cheating. Adding these behaviors will greatly reduce the model generalization. We must remove the behaviors with leaked labels, retrain the model, and discover the true behavior pattern of player cheating. The game cheating detection model recognizes normal pictures as cheating pictures in Fig. 10. We analyzed the explanations of the image explainer and found that the chat content and kill records of normal players are very similar to the plug-in elements. At the same time, the red crosshairs of the players are similar to the plug-in elements, which led us to mistakenly identify normal players as cheating players. By introducing more similar negative samples in the training set, we quickly improved the accuracy of the model and reduced the error blocking of the online model.

C. Model Compression and Comparison

Fig. 11 demonstrates the prediction comparison using the same classifier for input with top-k features selected by different explainers. As can be seen in Fig. 11(a), the predictions with portrait features selected by portrait explainers are better than those selected from random samples, among which XGBoost based explanation results showed the most outstanding performance. We also found that for game cheating detection, we only need 80 features to achieve near-optimal performance, and for player churn prediction, we merely need 60 of them. Fig. 11(b) illustrates the similar conclusion for the behavior view, among which Transformer based explanation results

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.

Fig. 11. (a)–(b)Prediction for input with top-k features. (a) GCD with portraits (XGBoost). (b) PCP with behaviors (Transformer).

showed the most prominent performance. We also found that we only need 600 behaviors to achieve nearly optimal performance for player churn prediction. Through the above feature selection based on model explanations, we have greatly compressed our model and improved the efficiency of model training and online inference.

VIII. Conclusion

In this article, we first introduced XAI in online games and proposed a GXAI workflow, which was evaluated with game cheating detection and player churn prediction. Extensive experiments show the accuracy of classification and the rationality of explanation. We have discovered and presented a lot of interesting and valuable findings from the model explanations. What is more, we implemented and deployed three practical applications in NetEase Games and received very positive reviews from user studies. We are working for the explainers of the multiview data based black-box models and expanding our work to more XAI applications in online games like explainable recommender systems, etc. We are also focusing on developing an XAI platform for explanations across AI lifecycle with pre and post deployment for AI models.

References

[1] J. Runge, P. Gao, F. Garcin, and B. Faltings, "Churn prediction for high-value players in casual social games," in *Proc. IEEE Conf. Comput. Intell. Games*, 2014, pp. 1–8.

[2] J. Tao *et al.*, "XAI-driven explainable multi-view game cheating detection," in *Proc. IEEE Conf. Games*, 2020, pp. 144–151.

[3] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[4] M. Moravčík *et al.*, "Deepstack: Expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508–513, 2017.

[5] D. Ye *et al.*, "Mastering complex control in MOBA games with deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 6672–6679.

[6] L. Gong *et al.*, "Optmatch: Optimized matchmaking via modeling the high-order interactions on the arena," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2300–2310.

[7] Y. Zheng *et al.*, ":Automatic online combat game testing using evolutionary deep reinforcement learning," in *Proc. IEEE/ACM Int. Conf. Automated Softw. Eng.*, 2019, pp. 772–784.

[8] T. Shi, Y. Yuan, C. Fan, Z. Zou, Z. Shi, and Y. Liu, "Face-to-parameter translation for game character auto-creation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 161–170.

[9] L. Liu, Y. Zheng, D. Tang, Y. Yuan, C. Fan, and K. Zhou, "Neuroskinning: Automatic skin binding for production characters with deep graph networks," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, 2019.

[10] B. Maher, "Can a video game company tame toxic behaviour?," *Nature*, vol. 531, no. 7596, pp. 568–571, 2016.

[11] J. Chen, Y. Liu, Z. Zhang, C. Fan, and Y. Ding, "Text-driven visual prosody generation for embodied conversational agents," in *Proc. 19th ACM Int. Conf. Intell. Virtual Agents*, 2019, pp. 108–110.

[12] J. Tao *et al.*, "MVAN: Multi-view attention networks for real money trading detection in online games," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2536–2546.

[13] Z.-J. Zhong, "The effects of collective MMORPG (massively multiplayer online role-playing games) play on gamers' online and offline social capital," *Comput. Hum. Behav.*, vol. 27, no. 6, pp. 2352–2363, 2011.

[14] M. L. Han, J. K. Park, and H. K. Kim, "Online game bot detection in FPS game," in *Proc. 18th Asia Pacific Symp. Intell. Evol. Syst.*, 2015, vol. 2, pp. 479–491.

[15] D. L. García, À. Nebot, and A. Vellido, "Intelligent data analysis approaches to churn as a business problem: A survey," *Knowl. Inf. Syst.*, vol. 51, no. 3, pp. 719–774, 2017.

[16] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," *Methods*, vol. 5, no. 13, p. 25, 2019.

[17] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6638–6648.

[18] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Stat*, vol. 1050, p. 4, 2018.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, pp. 1527–1535, 2018.

[22] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3145–3153.

[23] H. Liu, Q. Yin, and W. Y. Wang, "Towards explainable NLP: A generative explanation framework for text classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5570–5581.

[24] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 3319–3328.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

[27] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9240–9251.

[28] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang, "Graphlime: Local interpretable model explanations for graph neural networks," 2020, *arXiv:2001.06216*.

Authorized licensed use limited to: PUC MG. Downloaded on March 31,2024 at 14:10:32 UTC from IEEE Xplore. Restrictions apply.