Information Engineering Express
International Institute of Applied Informatics
2022, Vol. 8, No. 1, IEE316

An Agent-based Approach for Preventing and Defusing Toxic Behaviors on Team Competition Games

Kanji Watanabe *, Naoki Fukuta †

Abstract

Preventing toxic behaviors and defusing their negative effects in a team competition game is an important issue since it could cause serious problems as well as giving players poor experiences in their gaming. In this paper, we propose an approach to defuse the effects of toxic behaviors in a team-competition game and an agent-based framework to implement a mechanism which effectively defuses negative impacts of them as well as helping users to notice the meaning and impacts of certain toxic actions and avoid further chain-reactions from them.

Keywords: Toxic Behavior, Team Competition Game, Agent

1 Introduction

On the emergence of Internet-based communications, people have various opportunities to have communications on Reddit or other BBS services, Facebook and other SNS services, and Twitter and other microblogging services. These communications are often called "Computer Mediated Communication"[1]. There are emerging researches on the communications among players of online games[1], and consensus building support on these discussions[2].

Even on conversations and discussions on such computer mediated communications, sometimes these communications will heat up and thus they could cause conflicts among the members of them. In the context of online gaming, a kind of multi-player online battle arena games are incorporating such communications into their games to make better gaming experiences with other players. Although such competitive gaming designs make the gaming more attractive[3] [4], sometimes it also leads some players to make negative behaviors to other players called *toxic behaviors*. Furthermore, such toxic behaviors are observed even for non-verbal communications, which would make some positive effects to the performance of gaming when they are properly used[5]. In [1], an analysis has been presented to evaluate effects of such toxic behaviors to the gaming performance from the data obtained from crowdsourcing-based players' own reports.

In [1], it has been reported that such toxic behaviors may produce unwanted and poor gaming experiences as well as negative effects to the performance of the players in the

* Graduate School of Integrated Science and Technology, Shizuoka University
† College of Informatics, Academic Institute, Shizuoka University

game[1]. One of major difficulties on toxic behaviors in gaming is that, often it is not easy to clearly define what are toxic behaviors and this makes it difficult to predict and notice how a behavior could be a serious toxic behavior to the other players. This happens because of the diversity of their ethics, customs, and sometimes some of the players would apply some local rules which are only shared with a limited number of players.

In this paper, we propose an agent-based approach[1] to prevent and defuse negative impacts of such toxic behaviors in online gaming. We show a detailed design of our framework to implement such an agent by integrating machine learning based classifications of behaviors as well as giving context-aware actions for agents to give the players an *empathy* to them.

2 Background

2.1 User Support Agents and Effect of Empathy

There exists approaches to utilize software agents to support some complex works with humans. For example, in [7] and [8], they proposed a learning agent which could cooperate with both users and other agents. In [9], an approach has been presented to increase user's performance by using personalization of user support agents on a single-player video game. In [10] [11] [12] [13] and [14], an empathic virtual agent could contribute better performance of users in context of pedagogical tasks. However, to our best knowledge, there is little work on applying these approaches to defuse toxic behaviors in multiplayer gaming environments.

2.2 Effect and Problem of Toxic Behavior

In this paper, we define a *toxic behavior* as any negative action or behavior which can be seen in a kind of games which have some competitive contexts to other players. It is said that toxic behavior will make some negative effects of the players' gaming performance to both the player who had such toxic behaviors and other players who have seen such a situation. An important issue to avoid toxic behaviors is that sometimes it is difficult for the 'toxic' players to notice and understand their plays are toxic to other players due to the difference on their basic ethics and customs[15]. In this paper, to defuse the negative effects of such toxic behaviors and avoid chain-reactions of them to other players, we implement an interactive agent based on the players' behaviors in the game and help the users to notice the possible effects and meanings of behaviors from other players and give opportunities to interact the players each other to avoid misunderstandings of their behaviors to other players.

3 Design of Toxic Behavior Defusing Agents

3.1 Structure of Toxic Behavior Defusing Agents

The mechanism of our toxic behavior defusing agents can be seen as an extension of the approach proposed by Hasties[10] and we extended the approach to cover various players' models.

[1] Initial ideas about this work have been presented in [6].

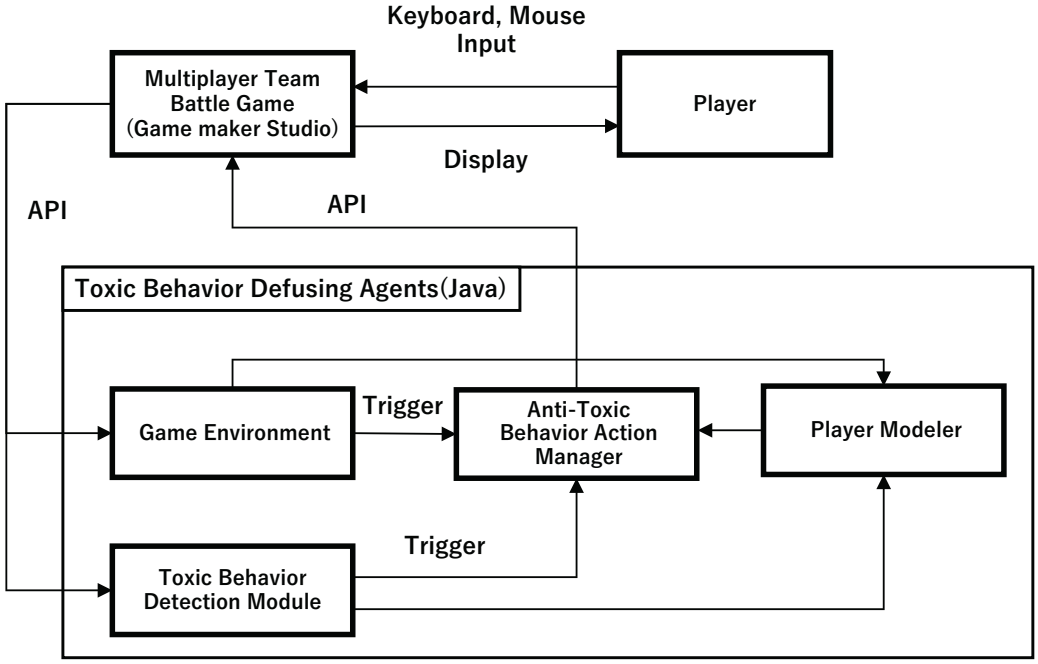

Figure 1: Structure of Toxic Behavior Defusing Agents

The structure of an entire system including our toxic behavior defusing agents is shown in figure1.

Our Toxic Behavior Defusing Agents consist of several modules, i.e., Toxic Behavior Detection Module, Player Modeler, Game Environment, and Anti-Toxic Behavior Action Manager.

Toxic Behavior Detection Module monitors player's ping, game information of environment, and text chat data to classify player's behaviors when weather each action is toxic behavior or not. Player Modeler predicts players' skills and mental models for agents that can make empathic actions based on their past events experienced with the players. Game Environment monitors its game environment of the by team competition game to allow agent to do better actions to the player. Anti-Toxic Behavior Action Manager monitors these modules' outputs and decides proper interactions to the player.

We allow the Toxic Behavior Defusing Agents to be at the outside of the competition game environment itself so that it can be applied to manage conflicts among players whose actions were done outside of the gaming environment (e.g., SNS, or on other games). Player can choose whether to use our agents and it allows storage of user model into player's local environment so player can control personal information.

3.2 Toxic Behavior Detection Module

Toxic Behavior Detection Module classifies player's behavior in the gaming whether it is toxic behavior or not by using the built-in APIs and allows agents to act according to the player's behaviors.

In this paper, to make the discussion simple, we would focus on an application for Mul-

tiplayer Online Battle Arena as the competition game and try to classify toxic behavior in the communications among team players on these types of games. As our initial analysis, we apply our method to the logs we have gathered from League of Legends and classify both Ping communications as non-verbal communications and text chats as verbal communications.

Toxic Behavior Detection Module detects player's malicious pings by using an SVM-based classifier, and player's malicious chats by Toxic N-gram[16] based classifier, respectively. This module monitors its communication data from the API and triggers actions. if toxic behavior were detected. Also these data are monitored on Player Modeler too. Figure3 shows an example of how our framework monitors players' actions on their system. As in the terminal logs, the system monitors the players' key actions and feeds these actions into the Toxic Behavior Detection Module.

3.2.1 Toxic Behavior Detection from Ping

To classify toxic behavior from pings in the gaming, we use Weka[17][1] as its implementation.

To confirm the applicability of our classifier to a real-world context, we examined the possible performance on the dataset obtained from match replays of League of Legends which included pings in the actions.

Table 1: Comparison result of classification possibility

Classifier	Precision	Recall	F-Measure
Decision Tree	0.913	0.914	0.907
RandomForest	0.943	0.938	0.933
Bayesian Network	0.965	0.963	0.961
SVM	0.962	0.963	0.962

We extracted 81 pings data (15 toxic pings and 66 non-toxic pings) manually from the data obtained from match replays of a gold (about top 40%) rank player's[2] *ranked* games. All annotations to the toxic behaviors have been done by the gold rank player who provided the match replays. Attributes that we used are game phase, kill count in the team, death count in the team, player's kill count, player's death count, number of times the ping was hit repeatedly, kind of the ping, place of the ping, whether the player is moving or not, and whether the player is alive or dead.

Machine learning algorithms we have applied to compose classifiers are Decision Tree (C4.5)[18], RandomForest[19], Bayesian Network[20], and SVM[21]. We evaluate their preliminary performances based on the results of the weighted average scores of Precision, Recall, and F-Measure in 10-fold cross validation. In Table1, we can see that the results on Bayesian Network and SVM are better than others.

3.3 Player Modeler

Player Modeler provides a prediction of the current condition to allow empathic actions of agents. Player Modeler uses data about player's pings and chats communications data pro-

[1] http://www.cs.waikato.ac.nz/ml/weka/

[2] One of the authors is a gold rank player from Q4 of 2016.

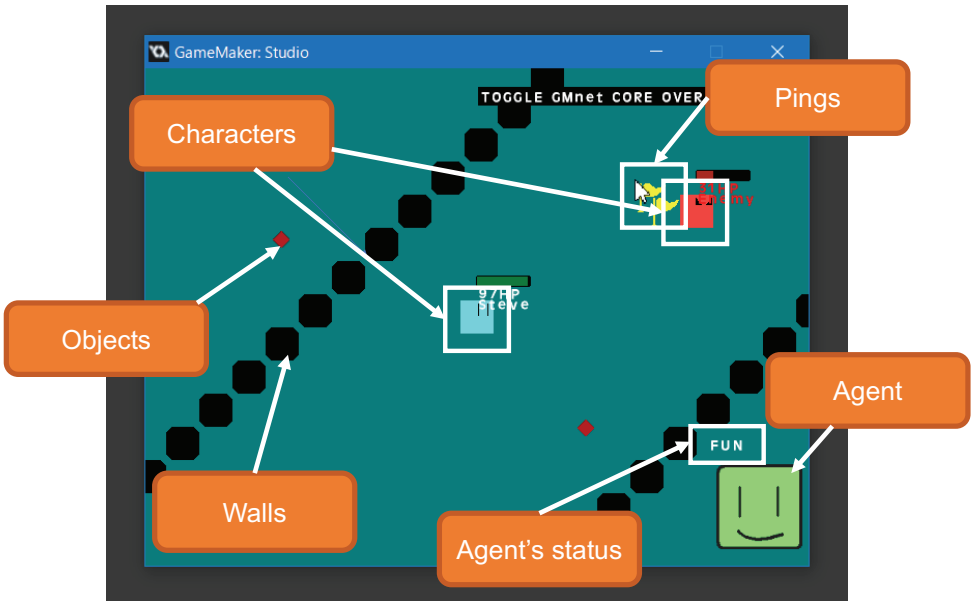

Figure 2: A screenshot of our prototype gaming environment

vided from Toxic Behavior Detection Module and predicts player's upset level. If player's frequency of communications is higher, Player modeler concludes that player get excited in gaming.

Furthermore, Player Modeler utilizes some contextual data of the game obtained from Game Environment and predicts player's emotional stability level. If player's team has a better chance of winning, Player Modeler predicts that the player is positive in gaming.

Player Modeler triggers associated actions if some changes on the player's emotional valance and upset level, and it also the past player's behaviors in gaming are used.

3.4 Game Environment

Game Environment is always monitored from associated modules using built-in APIs to trigger actions from the events happening at that time and it allows the agent to make some feedbacks to the user to encourage necessary communications within the team. Here, the data given from Game Environment includes transition of game phase, appearance of rare objects, and other data for the help of understanding the situation of the game.

3.5 Anti-Toxic Behavior Action Manager

Anti-Toxic Behavior Action Manager selects agent's anti-toxic behavior action based on from the given triggers Toxic Behavior Detection Module, Player Modeler, and Game Environment, and manages the entire actions to be done on the actual game environment.

To select right actions from there triggers, our agents allow empathic actions based on the given predictions of player's mental models obtained from Player Modeler to reduce negative effects on the gaming experience, and try to give the player an objective view of the events in the gaming environment if toxic behaviors are detected in the gaming environment.

For an example, warmly blocks and hides the toxic behaviors of another player in the gaming to avoid the player from becoming a toxic player by seeing such a negative or

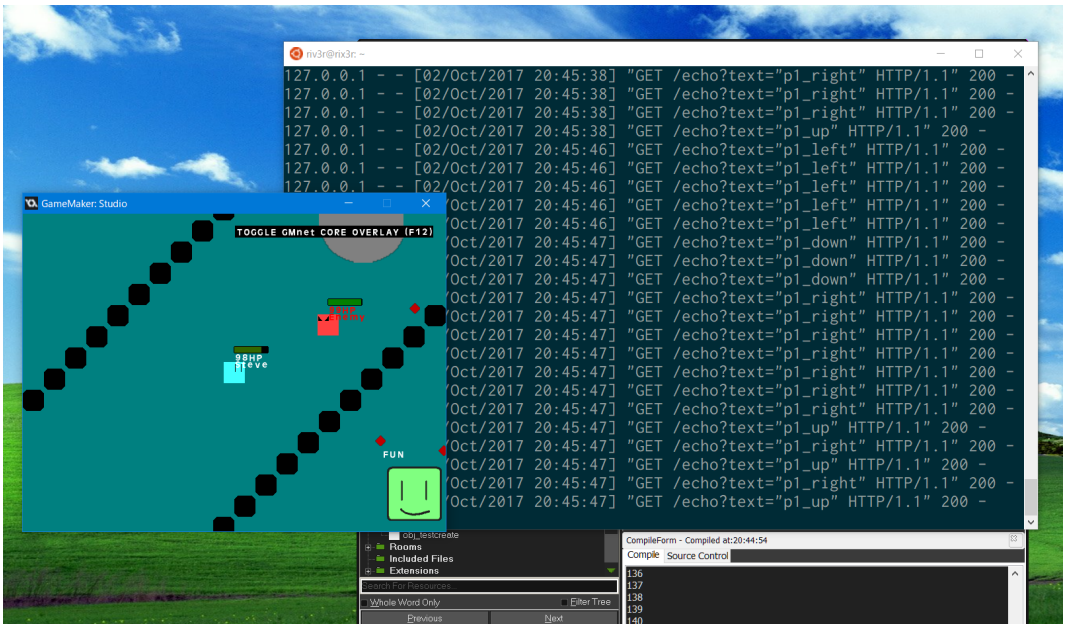

Figure 3: Overview of monitoring players' behaviors via the API

offensive behaviors in the gaming.

4 Implementation

We also implement a prototype of the gaming environment itself with the APIs that are necessary to run our agents. Our prototype gaming environment implements APIs to obtain gaming data that covers most of the features supported in the APIs in League of Legends[2]. We have been using Game Maker Studio[3] as its implementation platform.

Figure2 shows an example of prototype of our prototype implementation with its gaming environment. To keep the internal structure simple, currently our prototype gaming environment covers a 2 vs 2 team battle game and each team controls one character by two players (e.g., one can control movement and the other can control jumps and attacking actions). To win the game, players have to cooperate with the teammate.

Our agents act interactively to the player based on detected toxic behaviors and events happening. Figure4 shows an example of anti-toxic behavior on our agents. On the upper part of the figure, a player uses a danger ping in order to tell the one's teammate that we should run away from enemy because of the health of their operating character is very low. At this time, our agent judges that this behavior is harmless ordinary because the player uses a ping in a correct purpose within appropriate intervals. In the lower part of the figure, a player uses Mia (missing in area) pings to blame the one's teammate because of their character has been killed in spite of a caution. At this time, our agent judges that this behavior is toxic because the player uses ping for an inappropriate purpose. After that, the agent tries to defuse the detected toxic behaviors.

[2]http://na.leagueoflegends.com
[3]http://www.yoyogames.com/gamemaker

5 Discussion

5.1 Plan of Pattern Analysis Support to Toxic Players

In the previous sections, we discussed the way to realize toxic behavior defusing agents. To evaluate effective use of those agents, we need to expose the subjects to toxic behaviors on an evaluation experiment. There has an ethical issue of harming the subjects by conducting the experiment and it is difficult to verify the risk for the subjects. Furthermore, individual variance can be an issue when conducting a questionnaire survey. Therefore, we would not conduct experiments with subjects for this kind of evaluation but rather we would focus on defusing expression of toxic behavior.

Appropriate responses are different depending on whether the toxic player is looking forward to the act itself or if the emotions cannot be suppressed. Moreover, it is difficult for the toxic players to notice and understand their plays are toxic to other players due to the differences on their basic ethics and customs[15]. Therefore, by conducting a pattern analysis of the causes of toxic behavior, it will make possible to consider how to reduce the negative expression of toxic behavior for each type of the toxic player.

To conduct an analysis on the types of toxic players, there is an issue that deciding whether or not the specific behavior is toxic is highly subjective. We believe that this issue can be addressed by victims themselves deciding whether or not a game has been harmful to them. There is also a limitation to have the large amount of replay data that can be collected individually for the analysis. In this paper, we considered the feasibility of a platform as a basis for analyzing the type of toxic players, in order to study and apply the toxic behavior reduction methods according to the types of the toxic players.

Our pattern analysis platform to toxic players consists of several modules, i.e., Replay Editing and Annotation GUI, Replay Database, Replay Playback Application, and Synchronization Function.

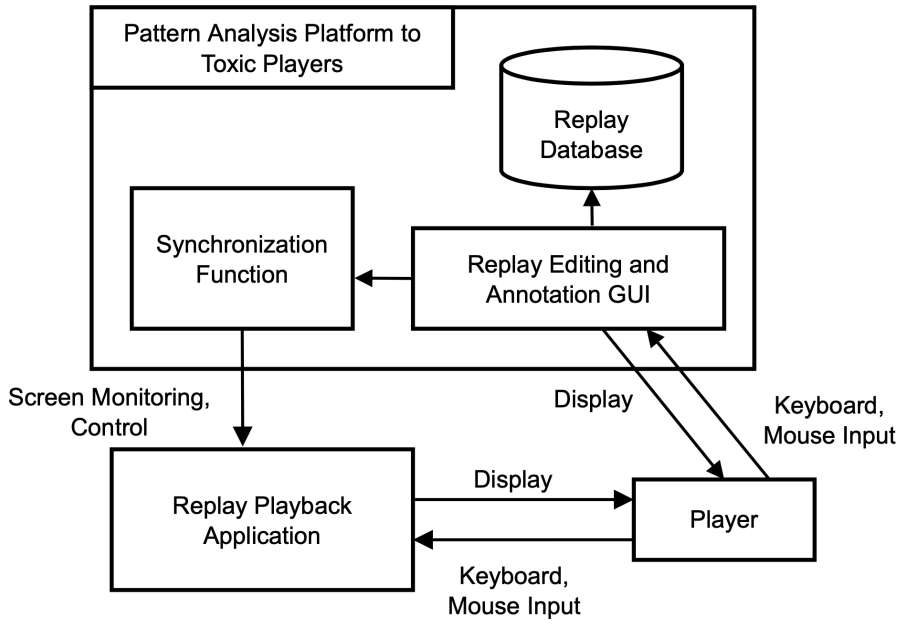

Figure 5: Structure of pattern analysis platform of toxic players

We are considering a platform that can support annotations by automatically extracting the scenes where toxic behavior occur, the scenes where misplays occur, and the scenes where communication is frequently performed from the replay data. Since it may take several hours per match to manually annotate all players' behaviors by referring to match replays, the automatic extraction of tens of seconds before and after a scene is expected to reduce the load of annotation work.

For the annotation tasks, we also need to consider the support for the classification of toxic players according to their types. It is assumed that the behavior of toxic players in the game will be varied depending on their basic ethics, customs, and the skill level of the players in the game. For example, by collecting match replays for each toxic player, it is expected that the replays can be used to support typological analysis of toxic players.

There is a probability of having mental loads when looking back on the player's own play (especially the one that made a mistake) that received the toxic behavior. When a person who is not received a toxic behavior refer to the replay, they can easily categorize it by annotating the scenes of the replay that should be paid attention to by the system. In addition, by using a filter, it may be possible to reduce the mental loads of the annotation.

5.2 Classification Possibility of Behavioral Intentions in Playing Games

When annotating a players' behaviors in match replay, it is expected that it will be one of the indicators for doing a typological analysis on the players' behaviors by considering not only their toxic behaviors but also their harmless ordinary communications during the game.

In communicating to other players, there are cases where the intention and the interpretation of a specific expression can be different depending on the context even with the same expression (e.g., [22]). For example, smiling emoji that follows a gentle sentence is completely different intentions from that of a same emoji that follows a sentence that makes a fool of the other person.

Even in playing games, due to the limitation of communication methods, there could have various intentions and interpretations even for the same action as an expression. Mia Ping, which appears in the form of a question mark in League of Legends, has multiple intents depending on the situation and context of the use during the game. For example, we could see the actual different usage patterns such as a pattern that used to alert allies in other locations when the facing enemy is no longer visible, a pattern that used to tell allies of the enemy's location when the enemy's location can be guessed, a pattern used with the intention of building consensus when there are multiple options, a pattern used to praise an ally's good play, and a pattern used to harass an ally.

In this paper, we categorized Mia Ping in League of Legends into four types: those used with original intentions, those that are not in the original intention but are used with the intention of making the game advantageous, those used to show harmless emotional expressions, and the toxic behaviors we focused in this paper. Then we examined the possibility of classifying these behaviors in the games.

To classify the intent of players' behaviors from pings in the gaming, we use Weka[17][1] and WekaDeeplearning4j[2] as the implementation platform of various ML algorithms.

To confirm the applicability of our classifiers to a real-world context, we examined the possible performance on the dataset obtained from match replays of League of Legends

[1] http://www.cs.waikato.ac.nz/ml/weka/

[2] https://deeplearning.cms.waikato.ac.nz/

which included pings in the actions.

We extracted 126 Mia pings data manually from the data obtained from match replays of a diamond (about top 0.1%) rank player's[3] *ranked* games. All annotations to the players' behaviors have been done by the diamond rank player who provided the match replays. Attributes that we used are elapsed time in the match, kill count in the team, death count in the team, number of times the ping was hit repeatedly, place of the ping, objects near the ping, whether the player is moving or not, whether the player is alive or dead, and the player's role.

Compared to the dataset used in the previous section, the main additions in this section are the skill level of the target players and the role of the pinged player. We can assume that the higher skilled players typically have a better understanding of the game and a better ability to overview the situation around them than the lower skilled players, therefore it is assumed that there are different causes of conflicts among the players.

Machine learning algorithms we have applied to compose classifiers are Decision Tree (C4.5)[18], RandomForest[19], Bayesian Network[20], SVM[21], and Convolutional Neural Network[23]. For the Convolutional Neural Network, we performed a grid search for the number of units per layer (32,64,128,256,512) and the dense layer (2,4,6,8), and set the best hyperparameters.

We evaluate their preliminary performances based on the results of the weighted average scores of Precision, Recall, and F-Measure in 10-fold cross validation. In Table2, we can see that the results on C4.5 are better than others, as well as the all classification algorithms obtained of 0.75 or higher in F-Measure.

Table 2: Comparison result of multiclass classification possibility

Classifier	Precision	Recall	F-Measure
Decision Tree(C4.5)	0.846	0.849	0.840
RandomForest	0.762	0.778	0.760
Bayesian Network	0.780	0.786	0.782
SVM	0.748	0.778	0.751
CNN (Dense layers:6, Units:64)	0.774	0.786	0.775

Table 3: Hyperparameter search results (F-Measure)

Units \ Dense layers	2	4	6	8
36	0.717	0.736	0.721	0.721
64	0.736	0.730	0.775	0.740
128	0.718	0.738	0.735	0.716
256	0.718	0.726	0.706	0.657
512	0.695	0.728	0.724	

Comparing this result with the comparison of two-classes toxic behavior classification performances shown in the previous section, the possible reasons why the classification performance of C4.5 is higher than that of other classification algorithms are: the player's role

[3]One of the authors is a diamond rank player from Q4 of 2021.

may have contributed significantly to the classification by the decision tree, communication in matches of higher skilled players may have been more well organized than that of lower skilled players, or overfitted model was generated. To confirm the first assumption, we prepared a classifier by C4.5 excluding the attribute of the player's role in the dataset and the obtained F-Measure was 0.827. Therefore, it was confirmed that the same trend can be seen even if there is no attribute of the player's role in the dataset. However, the result still has limitations on the absolute classification performance to be used in the applications.

6 Conclusion

In this paper, we proposed an approach to defuse the effects of toxic behaviors in a team-competition game. Also we proposed an agent-based framework to implement a mechanism which effectively defuses negative impacts of them as well as helping users to notice the meaning and impacts of certain toxic actions and avoid further chain-reactions from them. We also showed our prototype gaming environment with the APIs that are necessary to run our agents. Further analysis and improvement could be done based on the proposed framework.

Furthermore, this paper focuses on the reduction of toxic behavior itself as an extension of the toxic behavior defusing agents in multiplayer games, and describes pattern analysis support to toxic players in order to consider how to reduce toxic behavior according to the type of toxic players. To consider the possibility of classifying the intentions of players' behaviors, we annotated Mia ping in highly skilled group matches, which the first author actually played in League of Legends, and we then compared the classification performances on several machine learning algorithms on the annotated data. As a result of the comparison, the classification performance of C4.5 tends to be higher than that of other classification algorithms. One possible reason for this is that communication in the matches of higher skilled players may have been more well organized than that on the low skilled players.

References

[1] S. H. Haewoon Kwak, Jeremy Blackburn, "Exploring cyberbullying and other toxic behavior in team competition online games," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 3739–3748.

[2] T. Ito, Y. Imi, M. Sato, T. Ito, and E. Hideshima, "Incentive mechanism for managing large-scale internet-based discussions on collagree," in *Collective Intelligence*, 2015.

[3] C. K. Peter Vorderer, Tilo Hartmann, "Explaining the enjoyment of playing video games: The role of competition," in *Proceedings of the second international conference on Entertainment computing*, 2003, pp. 1–9.

[4] K. Sugiura and N. Fukuta, "A multiagent reinforcement learning approach for cooperative game playing with users on a sugorokulike board game," *International Journal of Information Technology*, vol. 22, no. 4, pp. 1–16, 2016.

[5] J. C. Alex Leavitt, Brian C. Keegan, "Ping to win? non-verbal communication and team performance in competitive online multiplayer games," in *Proceedings of the*

2016 CHI Conference on Human Factors in Computing Systems, 2016, pp. 4337–4350.

[6] K. Watanabe and N. Fukuta, "Toward empathic agents for defusing toxic behaviors on team competition games," in *Proc. 6th IIAI International Congress on Advanced Applied Informatics (IIAI AAI2017 / SCAI2017)*, 2017.

[7] S. Oishi and N. Fukuta, "A cooperative task execution mechanism for personal assistant agents using ability ontology," in *International Conference on Web Intelligence (WI2016)*, 2016, pp. 664–667.

[8] ——, "Toward a negotiation-based cooperation mechanism for user assistance agents and humans," in *Proc. the 10th International Workshop on Agent-based Complex Automated Negotiations (ACAN2017)*, 2017.

[9] A. Shvartzon, A. Azaria, S. Kraus, C. V. Goldman, J. Meyer, and O. Tsimhoni, "Personalized alert agent for optimal user performance," in *Association for the Advancement of Artificial Intelligence (AAAI2016)*, 2016, pp. 15–20.

[10] H. Hastie, M. Y. Lim, S. Janarthanam, A. Deshmukh, R. Aylett, M. E. Foster, and L. Hall, "I remember you! interaction with memory for an empathic virtual robotic tutor," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS2016)*, 2016, pp. 931–939.

[11] A. Deshmukh, A. Jones, S. Janarthanam, H. Hastie, T. Ribeiro, R. Aylett, A. Paiva, G. Castellano, M. E. Foster, and L. J. C. et al, "An empathic robotic tutor in a map application," in *Proc. of the 2015 International Conference on Autonomous Agents & Multiagent Systems (AAMAS2015)*, 2015, pp. 1923–1924.

[12] W. Burleson, "Affective learning companions: Strategies for empathetic agents with real-time multimodal affective sensing to foster meta-cognitive and meta-affective approaches to learning, motivation, and perseverance," in *Ph.D. dissertation, Massachusetts Institute of Technology, Massachusetts*, 2006.

[13] S. Janarthanam, H. Hastie, A. Deshmukh, R. Aylett, and M. E. Foster, "A reusable interaction management module: Use case for empathic robotic tutoring," in *Proceedings of SemDial 2015 (goDIAL): The 19th Workshop on the Semantics and Pragmatics of Dialogue*, 2015.

[14] S. Janarthanam, H. Hastie, A. Deshmukh, and R. Aylett, "Towards a serious game playing empathic robotic tutorial dialogue system," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction (HRI '14)*, 2014, pp. 180–181.

[15] T. Chesney, I. Coyne, B. Logan, and N. Madden, "Griefing in virtual worlds: causes, casualties and coping strategies," in *Information Systems Journal*, vol. 19, no. 6, 2009, pp. 525–548.

[16] M. Märtens, S. Shen, A. Iosup, and F. Kuipers, "Toxicity detection in multiplayer online games," in *Proceedings of the 2015 International Workshop on Network and Systems Support for Games*, 2015, pp. 1–6.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[18] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[19] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1995.

[20] J. Pearl, "Bayesian networks: A model of self-activated memory for evidential reasoning," in *Proceedings of the 7th Conference of the Cognitive Science Society*, 1985, pp. 329–334.

[21] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[22] T. Kawai and N. Fukuta, "Do you mean i was wrong? a preliminary approach on a graph-based framework for suggesting alternate interpretations on japanese conversations," in *International Symposium on Electrical and Computer Engineering Track, 17th International Conference on Quality in Research (QiR2021)*, 2021, pp. 135–140.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

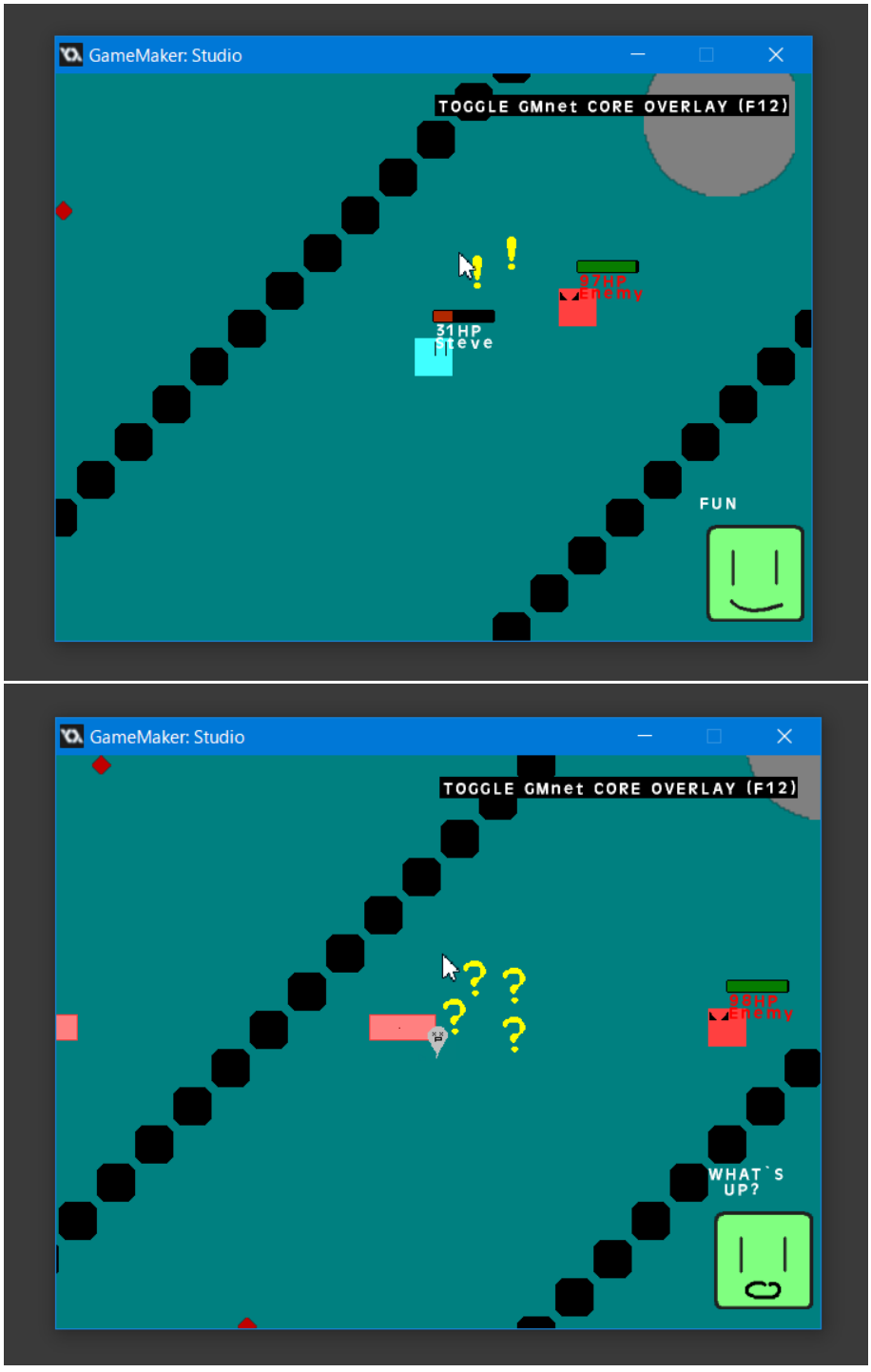

Figure 4: An example of agent's behaviors in harmless ordinary gaming (upper) and some toxic behaviors detected (lower)

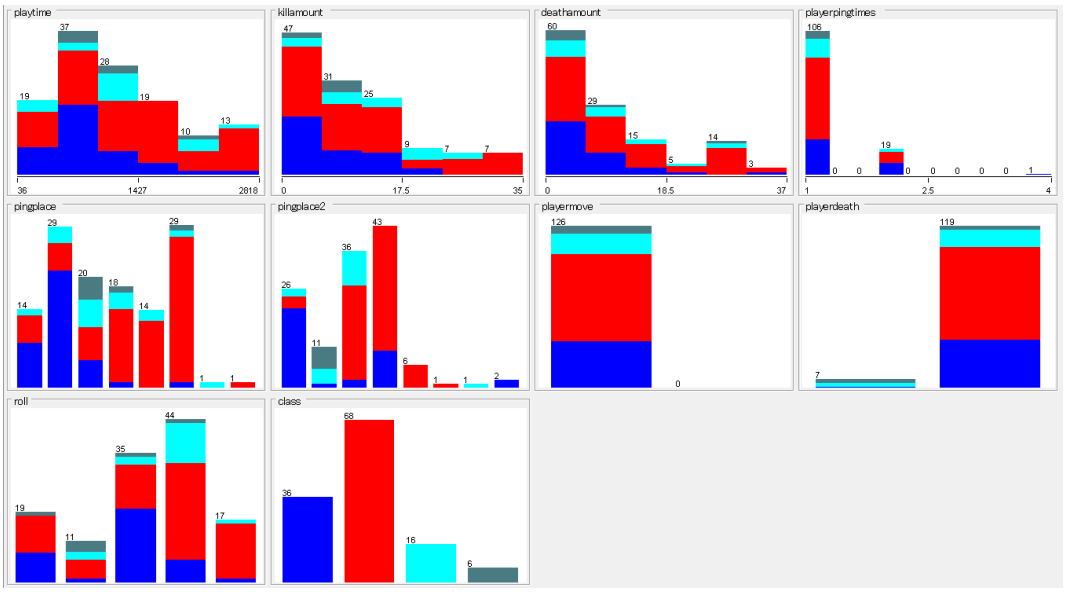

Figure 6: Distribution of attribute in training data