Using Data Augmentation and Machine Learning for Identifying the Correct Level of Difficulty in Games

André Gomes da Silva[1], **Fernando Santos**[2],
Augusto Baffa[2], **Jose Ricardo da Silva Junior**[1]

[1] Digital Games Technology Program – Federal Institute of Rio de Janeiro (IFRJ)
Engenheiro Paulo de Frontin, RJ – Brazil

[2]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro, RJ – Brasil

andre.gs.1618@gmail.com, fernandobsantos@aluno.puc-rio.br,
abaffa@inf.puc-rio.br, jose.junior@ifrj.edu.br

***Abstract. Introduction**: Dynamic Difficulty Adjustment (DDA) aims to balance player experience by adapting game challenge to individual abilities. Traditional approaches rely either on static difficulty selection or data-intensive adaptive models, both of which present limitations regarding player self-assessment and data availability. **Objective:** This work proposes a machine learning approach to identify the most appropriate difficulty level for games based on player performance metrics and subjective perceptions, while addressing the challenge of limited training data. **Methodology or Steps**: We model the problem as a three-class classification task (Increase, Keep, Decrease difficulty) using Logistic Regression. We adopt a CTGAN-based data augmentation approach to generate synthetic player data, thereby addressing data scarcity. The approach is evaluated using gameplay data from Doom II sessions, incorporating both objective performance metrics and players' perceived difficulty. **Results:** Experimental results show that models trained with synthetic data improve predictive performance compared to models trained only on real data, achieving higher F1-scores and better generalization. The findings suggest that synthetic data can enhance learning in low-data scenarios, although a domain gap between synthetic and real data persists, particularly when distinguishing between similar behavioral classes.*

***Keywords** Dynamic Difficulty Adjustment, Player Modeling, Support Vector Classification, Data Augmentation, Synthetic Data, Game Analytics*

1. Introduction

One of the most important aspects of developing a game is balance, to ensure a fair experience for the player [Schell 2008]. Balancing involves defining the game's difficulty by presenting challenges appropriate to a specific range of abilities. If challenges are too easy or too hard for a player, boredom or anxiety may arise, leading to churn. Besides that, challenges that closely match the abilities of a player and their evolution are one of the prerequisites to achieve the state of flow [Csikszentmihalyi e Csikszentmihalyi 1992].

During development, game balancing is typically performed using a develop-test-develop approach [Davis et al. 2005], in which game parameters are adjusted and

tested. Another common practice is beta testing [Davis et al. 2005], which is used as the game nears release. The beta testing approach consists of collecting feedback from a set of participants (beta testers) who play the game to identify technical issues, bugs, or gameplay imbalances. Normally, these beta testers are volunteers who play an early pre-release build of the game and provide valuable information on technical issues. Additionally, feedback collected during this process is used to adjust the game's difficulty in an artisanal manner, resulting in a more precise difficulty curve. In this approach, a set of static difficulty levels is typically presented to the player, and the player must choose the one they deem most appropriate for their abilities while playing the game.

Another common approach is to dynamically adjust the game's difficulty based on the player's current skill level, which requires monitoring players' actions and performance throughout the game. Such approaches are referred to in the literature as "dynamic game balancing" (DGB) and "dynamic difficulty adjustment" (DDA) [Hunicke 2005]. Several approaches for performing such dynamic balancing have been proposed, including prediction models [Hunicke 2005, Hawkins et al. 2012, Missura e Gärtner 2011], Evolutionary Fuzzy Cognitive Maps [Pérez et al. 2015], neuro-evolution [Olesen et al. 2008], challenge-based prediction [Mourato et al. 2014], and game scenario adaptation [Rietveld et al. 2014].

Both letting the player select a difficulty level from a predefined set of difficulty curves and using DDA approaches have drawbacks. By allowing the player to select a difficulty level, we rely on the player to be assertive in choosing a level that closely matches their abilities, a process that requires experience to make a fair judgment. Besides that, not all games in the same genre present the same challenge curve, which may affect the applicability of prior experience to future games. For DDA mechanisms, a large amount of data and model fine-tuning are necessary to improve precision in presenting users with challenges that increase in difficulty as their abilities grow over time. However, acquiring this dataset for training often imposes constraints, such as requiring a large number of game sessions from different participants to ensure data diversity. In addition, simply using the player's performance metrics to increase or decrease the level of difficulty, without further analysis, may lead to a situation where the player can exploit this to gain an advantage during the game [Hunicke 2005].

In this paper, we propose an approach that uses the player's perceived difficulty to identify the most appropriate level from a set of difficulty curves, achieving an F1-score up to 0.73 for the *Keep* class. Our approach employs a Logistic Regression machine learning model to analyze how a set of performance metrics relates to a player's desire to *Keep*, *Decrease*, or *Increase* the difficulty level. To address data scarcity, we employ a data augmentation mechanism to generate synthetic data, with a similarity score of 57.13% to real data. In summary, the paper has the following contributions:

- A machine learning model based on Logistic Regression for indicating the correct level of difficulty for a player;
- Data augmentation and evaluation of synthetic data generation regarding preservation of statistical properties in relation to real data; and
- Comparison of different models while using synthetic data and how they can represent real data.

This paper is organized as follows: Section 2 presents related work, while Section

3 provides background on synthetic data generation. Section 4 presents the materials and methods used for the experiment, and Section 5 presents our evaluation. Finally, Section 6 concludes this paper and points to future work.

2. Related Work

Dynamic Difficulty Adjustment (DDA) has shifted from heuristic rules to data-driven models that personalize challenge based on player behavior and subjective experience [Tran et al. 2025]. Early approaches, such as AI Director dynamic scaling in Left 4 Dead, adjusted enemy behavior using performance heuristics but lacked explicit modeling of player perception. More recent work emphasizes perceived difficulty as a key target using physiological signals to infer anxiety level [Liu et al. 2009], mental workload [Afergan et al. 2014], and emotional arousal [Rosa et al. 2021], demonstrating feasibility but relying on specialized hardware. Similarly, experience-driven adaptation surveys highlight the gap between objective performance metrics and subjective players' desire for changes in difficulty [Lopes et al. 2025].

Supervised learning with performance telemetry has shown promise in predicting difficulty without physiological sensors. Bicalho et al. [Bicalho et al. 2024] developed a Unity plugin that uses supervised classifiers, such as SVMs, for real-time telemetry-based adjustments. Figueira et al. [Figueira et al. 2018] presented BinG, a framework for dynamic game balancing that analyzes a player's session history via a provenance graph. Melhart et al. [Melhart et al. 2019] used clustering and regression on Ghost Recon telemetry to infer players' motivation from gameplay features. However, these studies focus on binary or continuous skill estimates, and they often suffer from **data scarcity**—especially for underrepresented player profiles (e.g., novices or quitters). This imbalance biases classifiers toward majority classes, reducing reliability for real-time DDA [Kristensen et al. 2022].

To address data scarcity, data augmentation has emerged in the gaming industry. Data augmentation techniques such as the Synthetic Minority Oversampling Technique (SMOTE) have been employed to mitigate class imbalance and address small sample sizes. Sifa et al. [Sifa et al. 2018] successfully used SMOTE to improve predictions of high-value users in freemium games by augmenting minority-class behavioral data. More recently, Sifa and Yang [Sifa e Yang 2023] conducted a comparative analysis of data augmentation approaches, including SMOTE, Variational Autoencoders (VAEs), and Generative Adversarial Networks (GANs), and found that these techniques significantly improve model generalization for retention prediction tasks. However, evaluating the quality of the generated data remains a critical concern. Most current research in the games domain evaluates synthetic data using "ML utility"—that is, whether the augmented data improves the performance of a downstream classifier [Sifa e Yang 2023, Pelling e Gardner 2019]. For example, Li et al. [Li et al. 2010] generated synthetic training data using Monte Carlo Tree Search (MCTS) to train an artificial neural network for DDA and validated the approach by evaluating the network's performance in a testbed. However, fewer studies focus on "statistical preservation"—the degree to which synthetic data maintain the statistical properties and distributions of the original real-world dataset. The current paper addresses this gap by evaluating how well synthetic data preserves these properties.

Our work bridges these gaps by: (1) proposing an Logistic Regression-based model that directly maps performance metrics to three-class perceived difficulty preferences (keep/decrease/increase); (2) introducing a targeted data augmentation mechanism for generating synthetic player traces, with evaluation of statistical preservation (distribution, correlation, variance); and (3) comparing Logistic Regression models under synthetic vs. real data to assess representativeness. Unlike prior studies, we prioritize player-reported desire over inferred skill, enabling DDA that aligns with subjective experience while remaining robust to data scarcity.

3. Data Augmentation

In supervised machine learning, a model's performance is typically directly related to the amount of training data available. In the last year, a majority of video-game databases have become available, ranging from specific experiments on model affects [Melhart et al. 2022] and player engagement [Rashed et al. 2025] to initiatives aimed at collecting and sharing game datasets for different purposes [Melhart et al. 2022]. However, in most cases, these databases do not meet the experiment's requirements and often lack important information.

One solution to data scarcity is data augmentation, which uses real data to learn patterns and generate new, statistically valid data. To perform data augmentation and generate synthetic data, we employed the Synthetic Data Vault (SDV)[Patki et al. 2016] framework, which provides tools for modeling and generating synthetic tabular data. Specifically, we used the CTGAN synthesizer, a generative model based on Conditional Generative Adversarial Networks, designed to handle mixed data attributes. The CTGAN model is built on the adversarial training paradigm introduced by Goodfellow [Goodfellow et al. 2020], in which two neural networks are trained simultaneously: a generator G and a discriminator D. The generator aims to produce synthetic samples that resemble real data, while the discriminator attempts to distinguish between real and generated samples. The training objective of a standard GAN can be formulated as a minimax optimization, according to Equation 1

$$\min_G \max_D V(D,G) = \mathbb{E}x \sim pdata(x)[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))], \quad (1)$$

where x represents samples drawn from the real data distribution p_{data} and z represents a noise vector sampled from a latent distribution p_z. The generator transforms the noise vector into a synthetic sample $G(z)$, while the discriminator outputs the probability that a given sample is real.

In CTGAN, conditional generation is introduced to better model tabular datasets with heterogeneous distributions. Instead of generating samples solely from random noise, the generator receives both a latent vector z and a conditional vector c representing a specific discrete attribute value. The generator therefore learns a conditional distribution $G(z, c)$ where c guides the generation process to ensure that discrete variables follow realistic distributions. The discriminator is also conditioned on the same vector, learning to evaluate whether the generated sample satisfies the conditional constraint. This conditioning strategy helps address the common problem of imbalanced categorical features in tabular datasets.

Our ground truth during training is based on the player's perception of the level of difficulty at the end of the game session, including whether to keep the previously selected difficulty or adjust it up or down. By doing so, we shift responsibility for determining the appropriate level of difficulty to the player after they play at the difficulty level they selected at the start of the session.

4. Methodology

To generate synthetic data, we used a real dataset to train a classification model to determine whether the player should stay at the selected difficulty level, decrease it, or increase it. For this purpose, we opted to use a video game with predefined difficulty levels, developed by recognized professionals in the video game industry and well-established in the market. Additionally, one criterion for selecting the game is the availability of the source code for modification, as changes would be necessary to enable data collection. Given that different game genres use distinct evaluation metrics due to their respective levels of complexity, we selected a game whose analysis would not be overly complex. After some examination, it was decided to use Doom II [id Software 1994], a 1994 first-person shooter developed by Id Software. It is a highly acclaimed and widely popular game, praised in reviews and consistently ranked among the top titles by various gaming publications. Doom II has five difficulty levels, which can be selected when starting a new game, as shown in Figure 1.

Figure 1. Available levels of difficulty in Doom II.

For our experiment, we selected the first three difficulty levels: *I'm too young to die* ($level_1$), *Hey, not too rough* ($level_2$), and *Hurt me plenty* ($level_3$), ranging from the easiest to the hardest. Among these levels, factors such as the number, placement, and aggressiveness of enemies, the damage the player receives, the availability of weapons and ammo, and modifiers are changed, making the game more difficult. From Doom II, nine performance metrics were collected, as described in Table 1. In addition, we incorporate

the player's perceived difficulty ($P_d \subset \{$*keep, decrease, increase*$\}$) by asking the player at the end of the session whether they would keep, decrease, or increase the difficulty level; this is used as the ground truth for the supervised machine learning model. At the end, our dataset D is composed of all the metrics in Table 1 and P_d. All these metrics are numeric, except for D_{dl}, which is categorical.

Table 1. Metrics collected during a game session.

Metric	Description
D_{dl}	Level of difficulty selected by the player.
D_{di}	Indicates the total damage dealt to enemies.
D_{dr}	Measures the total amount of damage endured by the player.
D_{nd}	Counts the number of times the player has died.
D_{mp}	The relation of the number of monsters defeated to the total number of monsters present in the stage, expressed as a percentage.
D_{ac}	Measures the player's aiming precision, expressed as a percentage. Calculated by analyzing the ratio of successful hits to total shots fired.
D_{tm}	Represents the total time in minutes the player took to pass each stage.
D_{dt}	Shows the total distance the player walked/ran throughout each stage.
D_{me}	Indicates the percentage of each stage visited by the player.

Considering all nine metrics in Table 1, we employed a CTGAN to train and generate synthetic data. No feature engineering was performed on the collected metrics; they were used as is. However, internally, CTGAN applies normalization to numerical features and one-hot encoding to categorical features. To avoid data leakage, we enforce a strict fold-based protocol in which the CTGAN and Z-score parameters are computed exclusively within each training fold, leaving the real test fold completely untouched. Additionally, due to imbalance in the target class, this fold is stratified.

To evaluate whether the player is at the right level of difficulty, we use Logistic Regression. Additionally, we use Support Vector Classification (SVC) as a baseline comparison due to its ability to accommodate more complex models via a radial basis function (RBF) kernel [Müller e Guido 2016]. Because both Logistic Regression and SVC are sensitive to numerical ranges, all numerical features in the dataset have undergone standardization (z-score normalization). However, we are not interested in determining which player is the best in the dataset, but rather in whether the player has selected the difficulty he judges to be best for his actual ability level. So, these metrics have been standardized within their respective difficulty groups ($n_{level1}, n_{level2}, n_{level3}$). Algorithm 1 describes the process of training and generating synthetic data as well as evaluating the metrics.

To collect data on game sessions, participants were invited, one at a time, to play the game in a private research laboratory over a one-month period. Before the game session started, participants were introduced to Doom II game controls, followed by a five-minute free play. After that, a 20-minute session took place during which the metrics specified in Table 1 were collected. At the end of the experiment, the participants answered an online form indicating if they would keep, decrease, or increase the level of difficulty. This study was approved by the Ethics Committee of Instituto Federal do

Algorithm 1: Stratified Cross-Validation Pipeline with CTGAN Data Augmentation.

Require: D_{real} (Original dataset with 29 records), K (Number of folds = 5), N (Synthetic samples to generate = 100)
Ensure: Mean $F1\text{-}Score$ for SVC and Logistic Regression models
Initialize `StratifiedKFold` splitter with $K \leftarrow 5$
Initialize empty lists: $f1\_svc\_total \leftarrow []$ and $f1\_lr\_total \leftarrow []$
for each $Fold_i$ from 1 to K **do**
  Partition D_{real} into D_{train_real} ($\approx 80\%$) and D_{test_real} ($\approx 20\%$)
  Train generative model `CTGAN` using **only** D_{train_real} {Target column configured as a discrete variable}
  $D_{synthetic} \leftarrow$ `CTGAN.sample`(N)
  $D_{train\_final} \leftarrow$ Concatenate$(D_{train_real}, D_{synthetic})$
  Apply One-Hot Encoding strictly to categorical attribute columns in D_{train_final}
  Compute normalization parameters (μ, σ) on D_{train_final} grouped by game difficulty levels
  Apply Z-Score normalization to D_{train_final} using the computed (μ, σ)
  Apply Z-Score normalization to D_{test_real} using the (μ, σ) parameters borrowed from the training
  Train Classifiers on normalized training data:
    Model$_{SVC}$.fit$(D_{train_final_normalized}, C = 1.0)$
    Model$_{LR}$.fit$(D_{train_final_normalized})$
  Evaluate models on the untouched $D_{test\_real\_normalized}$:
    $score\_svc \leftarrow$ Calculate_F1_Weighted$(D_{test_real_labels}, \text{Predictions}_{SVC})$
    $score\_lr \leftarrow$ Calculate_F1_Weighted$(D_{test_real_labels}, \text{Predictions}_{LR})$
  Append $score\_svc$ to $f1\_svc\_total$
  Append $score\_lr$ to $f1\_lr\_total$
end for
Compute final mean and standard deviation for $f1\_svc\_total$ and $f1\_lr\_total$
return Consolidated Final Metrics

Rio de Janeiro under protocol number 94602625.9.0000.5268. All participants provided informed consent prior to participating in this research.

5. Results

In total, we collected data from 31 participants' game sessions. Although the resulting dataset is relatively small, each record corresponds to a full supervised experimental session conducted individually in a controlled laboratory environment. Data collection was performed in person, with each session requiring approximately 30 minutes, including participant familiarization, gameplay, and the post-session questionnaire. Furthermore, the data collection period was limited to one month and conducted only on available working days, which constrained the total number of participants that could be recruited. Doom II presents 5 different difficulty levels, as shown in Figure 1, and participants were free to choose their preferred initial difficulty level. As a consequence, only two participants selected each of the two highest difficulty levels (levels 4 and 5), resulting in an insufficient number of observations for those levels. Therefore, results

were considered only across the first three difficulty levels ($n_{level_1} = 5$, $n_{level_2} = 8$, and $n_{level_3} = 16$). Approximately 93% of participants were aged 18 to 34 years, while the remaining were aged 35 to 54 years. Regarding playtime, 56% of participants report playing between 1 and 4 hours per day, 23% play more than 4 hours per day, and 13% play less than 1 hour per day.

Their perception of difficulty (P_d), answered at the end of the session, is shown in Table 2. According to this table, most participants ($n = 15$) felt they were not at the right level of difficulty, as they chose to decrease ($n = 7$) or increase ($n = 8$). Additionally, $level_3$ has the most participants choosing to increase or decrease the difficulty.

Table 2. Participants' perception of their selected level of difficulty.

Option	%	Difficulty	Total
Keep ($n = 14$)	48.28%	1	4 (28.57%)
		2	2 (14.29%)
		3	8 (57.14%)
Decrease ($n = 7$)	24.14%	1	1 (14.29%)
		2	2 (28.57%)
		3	4 (57.14%)
Increase ($n = 8$)	27.59%	2	4 (50.00%)
		3	4 (50.00%)

5.1. Data Augmentation

The synthetic data generation used a CTGAN model trained for 10 epochs on 29 previously collected real game sessions, producing 100 synthetic game sessions. All training and analysis were performed on Google Colab using an NVIDIA T4 GPU and a seed ($random\_state = 42$). The mean data quality for statistical similarity between the real and synthetic data was 67.58% for column shape and 46.68% for column pair trends, yielding an overall mean score of 57.13%.

5.2. Difficulty Classification Model

Initially, both Logistic Regression (LR) and the SVC model were trained only on real data ($n = 29$) to evaluate their precision in detecting the intention to keep, decrease, or increase the level of difficulty using performance metrics. In this scenario, $LR = 0.48 \pm 0.22$ and $SVC = 0.44 \pm 0.19$ for a mean weighted F1-score. Given the small size of the training data, a deterministic approach that maintains a constant level of difficulty would achieve a score of 0.48, since this class has the most participants, closely matching the model's score.

In the next experiment, 80% of the training dataset was augmented with 100 synthetic data records produced by CTGAN. In this scenario, $LR = 0.61 \pm 0.14$ and $SVC = 0.45 \pm 0.09$ for a mean weighted F1-score, demonstrating an increasing precision, especially for Logistic Regression, which demonstrated an increase of 0.13 in precision compared to the previous scenario trained only on real data. Table 3 presents the detailed classification performance for each class, including precision, recall, and F1-score for Logistic Regression model, collected during fold evaluation As can be observed, the

model is more assertive in detecting metric patterns in which the user opts to keep the level of difficulty ($F1 = 0.73$) followed by intention to decrease ($F1 = 0.67$), indicating a more distinct behavior of such gruoups. On the other hand, detecting metric patterns involving the user opting to increase the difficulty level achieved the lowest score ($F1 = 0.50$). One reason for this result is that experienced players prefer to play at a medium-to-hard difficulty level [Alexander et al. 2013], where behaviors are similar in staying at the selected difficulty versus increasing it.

Table 3. Synthetic classification model performance over real data.

	Classification Performance			
	Support	**Precision**	**Recall**	**F1-score**
Keep	14	0.69	0.79	0.73
Decrease	7	0.80	0.57	0.67
Increase	8	0.50	0.50	0.50

Figure 2 presents the multi-class evaluation of the proposed Logistic Regression model on real-world data. The confusion matrix (Figure 2(a)) shows that the model achieves good overall performance, particularly for the *Keep* class, which reaches 0.80 correct classification. However, a notable confusion is observed between the *Keep* and *Increase* classes, indicating overlapping regions in the feature space.

The Receiver Operating Characteristic (ROC) curves (Figure 2(b)) further confirm the model's discriminative capability, with AUC values of 0.93 for *Decrease*, 0.78 for *Increase*, and 0.71 for *Keep* classes. This suggests that the model is highly effective at distinguishing the *Decrease* class, while maintaining good, albeit imperfect, separability for the other classes. Considering the imbalanced classes in the dataset, the Precision-Recall curves (Figure 2(c)) provide additional insights, revealing that the *Increase* class exhibits greater variability in precision across different recall levels. This indicates that predictions for this class are more sensitive to the decision threshold, likely due to overlap with the *Keep* class.

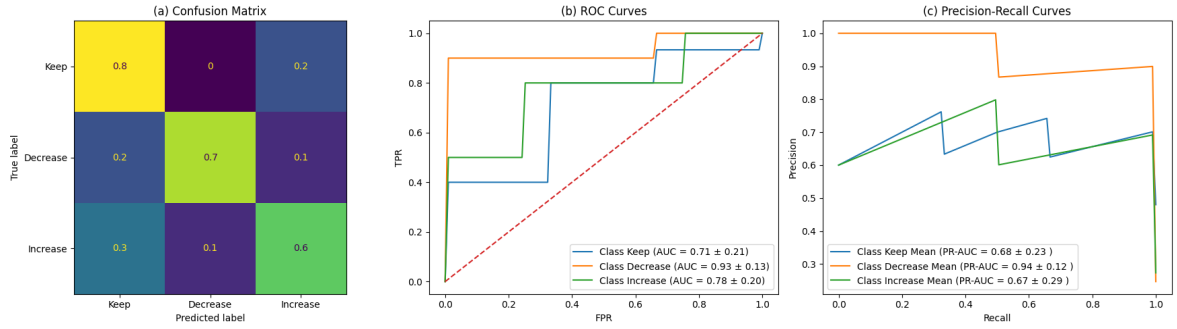

Figure 2. Multi-class performance evaluation of the Linear Regression model on real data. (a) Normalized confusion matrix, (b) One-vs-Rest ROC curves with AUC values, and (c) Precision-Recall curves for each class.

Overall, these results suggest that although the model generalizes reasonably well from synthetic to real data, a domain gap remains, primarily affecting its ability

to distinguish between similar behavioral classes, such as keeping the level of difficulty or increasing it.

6. Conclusion

This paper investigated the use of supervised machine learning combined with data augmentation techniques to estimate the appropriate level of difficulty in digital games based on player performance metrics and perceived difficulty. The problem was framed as a three-class classification task (*keep, decrease, or increase* difficulty), and linear regression and an SVC model (used as a baseline) were employed to relate gameplay telemetry to player-reported preferences.

Given the limited availability of real gameplay data, a CTGAN-based data augmentation strategy was adopted to generate synthetic samples, achieving an overall **similarity score of 57.13%** with real data. Experimental results indicate that models trained on augmented data achieved higher performance than those trained solely on real data, particularly in weighted F1-score and overall accuracy when evaluated on real-world samples. These findings suggest that synthetic data can partially mitigate data scarcity in player modeling scenarios, while preserving relevant statistical properties of the original dataset.

The results also open up opportunities for further improvement. Expanding the size and diversity of the real dataset may help enhance the model's generalization capability and provide a more comprehensive evaluation. Additionally, the partial overlap observed between some classes — particularly *Keep* and *Increase* — suggests that incorporating additional or more expressive performance metrics could further improve the model's ability to capture nuanced player behaviors. Finally, while the use of synthetic data has shown promising gains in predictive performance **(F1-score over 0.73 for the *Keep* class)**, exploring ways to further align real and generated data may yield even more robust results, especially in finer-grained classification scenarios.

In future work, we plan to address these limitations by collecting a larger, more diverse dataset, thereby enabling more robust statistical analyses, particularly regarding the similarity between synthetic and real data. We also intend to explore richer behavioral and contextual features, as well as alternative learning models and ensemble methods. In addition, the proposed prediction model will be integrated into a closed-loop dynamic difficulty adjustment (DDA) system, allowing real-time adaptation and longitudinal evaluation of player experience. Finally, we aim to further investigate the relationship between objective performance metrics and players' subjective perception, seeking a better alignment between adaptive systems and player-centered design.

7. Acknowledgments

The authors would like to thank CAPES, CNPq, and FAPERJ for the financial support.

References

Afergan, D., Peck, E. M., Solovey, E. T., Jenkins, A., Hincks, S. W., Brown, E. T., Chang, R., e Jacob, R. J. (2014). Dynamic difficulty using brain metrics of workload. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 3797–3806, New York, NY, USA. Association for Computing Machinery.

Alexander, J. T., Sear, J., e Oikonomou, A. (2013). An investigation of the effects of game difficulty on player enjoyment. *Entertainment Computing*, 4(1):53–62.

Bicalho, L. F., Baffa, A., e Feijó, B. (2024). A dynamic difficulty adjustment algorithm with generic player behavior classification unity plugin in single player games. In *Proceedings of the 22nd Brazilian Symposium on Games and Digital Entertainment*, SBGames '23, page 76–85, New York, NY, USA. Association for Computing Machinery.

Csikszentmihalyi, M. e Csikszentmihalyi, I. S. (1992). *Optimal Experience: Psychological Studies of Flow in Consciousness*. Cambridge University Press.

Davis, J. P., Steury, K., e Pagulayan, R. (2005). *A survey method for assessing perceptions of a game: The consumer playtest in game design*. Game Studies 5.

Figueira, F. M., Nascimento, L., da Silva Junior, J., Kohwalter, T., Murta, L., e Clua, E. (2018). Bing: A framework for dynamic game balancing using provenance. In *2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pages 57–5709.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., e Bengio, Y. (2020). Generative adversarial networks. *Commun. ACM*, 63(11):139–144.

Hawkins, G., Nesbitt, K., e Brown, S. (2012). Dynamic difficulty balancing for cautious players and risk takers. *Int. J. Comput. Games Technol.*, 2012:3:3–3:3.

Hunicke, R. (2005). The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '05, pages 429–433, New York, NY, USA. ACM.

id Software (1994). Doom II: Hell on earth. MS-DOS. Jogo eletrônico.

Kristensen, J. T., Guckelsberger, C., Burelli, P., e Hämäläinen, P. (2022). Personalized game difficulty prediction using factorization machines. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA. Association for Computing Machinery.

Li, X., He, S., Dong, Y., Liu, Q., Liu, X., Fu, Y., Shi, Z., e Huang, W. (2010). To create dda by the approach of ann from uct-created data. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, volume 8, pages V8–475–V8–478.

Liu, C., Agrawal, P., Sarkar, N., e Chen, S. (2009). Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. *International Journal of Human–Computer Interaction*, 25(6):506–529.

Lopes, P., Fachada, N., e Fonseca, M. (2025). Closing the loop: A systematic review of experience-driven game adaptation.

Melhart, D., Azadvar, A., Canossa, A., Liapis, A., e Yannakakis, G. N. (2019). Your gameplay says it all: Modelling motivation in tom clancy's the division. In *2019 IEEE Conference on Games (CoG)*, pages 1–8.

Melhart, D., Liapis, A., e Yannakakis, G. N. (2022). The arousal video game annotation (again) dataset. *IEEE Transactions on Affective Computing*, 13(4):2171–2184.

Missura, O. e Gärtner, T. (2011). Predicting dynamic difficulty. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., e Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2007–2015. Curran Associates, Inc.

Mourato, F., Birra, F., e dos Santos, M. P. (2014). Difficulty in action based challenges: Success prediction, players' strategies and profiling. In *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology*, ACE '14, pages 9:1–9:10, New York, NY, USA. ACM.

Müller, A. C. e Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Inc.

Olesen, J. K., Yannakakis, G. N., e Hallam, J. (2008). Real-time challenge balance in an rts game using rtneat. In *2008 IEEE Symposium On Computational Intelligence and Games*, pages 87–94.

Patki, N., Wedge, R., e Veeramachaneni, K. (2016). The synthetic data vault. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410.

Pelling, C. e Gardner, H. (2019). Two human-like imitation-learning bots with probabilistic behaviors. In *2019 IEEE Conference on Games (CoG)*, pages 1–7.

Pérez, L. J. F., Calla, L. A. R., Valente, L., Montenegro, A. A., e Clua, E. W. G. (2015). Dynamic game difficulty balancing in real time using evolutionary fuzzy cognitive maps. In *2015 14th Brazilian Symposium on Computer Games and Digital Entertainment*, pages 24–32.

Rashed, A., Shirmohammadi, S., e Hefeeda, M. (2025). Descriptor: Multimodal dataset for player engagement analysis in video games (multipeng). *IEEE Data Descriptions*, 2:17–25.

Rietveld, A., Bakkes, S., e Roijers, D. (2014). Circuit-adaptive challenge balancing in racing games. In *2014 IEEE Games Media Entertainment*, pages 1–8.

Rosa, M. P. C., dos Santos, E. A., de Moraes, I. L. R., e Silva, T. B. P., Sarmet, M. M., Castanho, C. D., e Jacobi, R. P. (2021). Dynamic difficulty adjustment using performance and affective data in a platform game. In Stephanidis, C., Soares, M. M., Rosenzweig, E., Marcus, A., Yamamoto, S., Mori, H., Rau, P.-L. P., Meiselwitz, G., Fang, X., e Moallem, A., editors, *HCI International 2021 - Late Breaking Papers: Design and User Experience*, pages 367–386, Cham. Springer International Publishing.

Schell, J. (2008). *The Art of Game Design: A Book of Lenses*. Morgan Kaufmann Publishers.

Sifa, R., Runge, J., Bauckhage, C., e Klapper, D. (2018). Customer lifetime value prediction in non-contractual freemium settings: Chasing high-value users using deep neural networks and smote.

Sifa, R. e Yang, E. (2023). A comparative analysis of data augmentation approaches for improved minority behavior detection in digital games. In *2023 IEEE International Conference on Big Data (BigData)*, pages 6275–6278.

Tran, T. D., Heinzig, M., Langner, H., e Ritter, M. (2025). Towards adaptive difficulty and personalized player experience: A data-driven approach combining game analytics, sensor data, and personality traits. In *Proceedings of the Mensch Und Computer 2025*, MuC '25, page 29–46, New York, NY, USA. Association for Computing Machinery.