Predicting the Number of Video Game Players on the Steam Platform Using Machine Learning and Time Lagged Features

Gregorius Henry Wirawan, Gede Putra Kusuma
Computer Science Department-BINUS Graduate Program-Master of Computer Science,
Bina Nusantara University, Jakarta 11480, Indonesia

***Abstract*—Predicting player count can provide game developers with valuable insights into players' behavior and trends on the game population, helping with strategic decision-making. Therefore, it is important for the prediction to be as accurate as possible. Using the game's metadata can help with predicting accuracy, but they stay the same most of the time and do not have enough temporal context. This study explores the use of machine learning with lagged features on top of using metadata and aims to improve accuracy in predicting daily player count, using data from top 100 games from Steam, one of the biggest game distribution platforms. Several combinations of feature selection methods and machine learning models were tested to find which one has the best performance. Experiments on a dataset from multiple games show that Random Forest model combined with Pearson's Correlation Feature Selection gives the best result, with R^2 score of 0.9943, average R^2 score above 0.9 across all combinations.**

Keywords*—Video games; regression method; feature selection; time series forecasting; machine learning

I. INTRODUCTION

The video game industry has seen a massive growth over the past few years, especially during the COVID-19 pandemic, when people were encouraged to stay at home, increasing gaming activity. The increased gaming activity was due to either stress relief [1], seeking social interactions [2], or having no other activity to do at home [3]. Steam is one of the rapidly growing game distribution platforms and makes a major contribution in the growth of the industry, along with the transition from physical distribution of games to digital distribution in the form of licenses, and along with online social networking services for gamers [4].

With more than 50,000 video games available from various developers and publishers, Steam gives gamers the liberty to buy and play their favorite video games and share their captured moments with their friends on the platform. Gamers can also leave their impressions and share their opinions about the game they play through a review on the game's store page [5]. All the things happening on the platform will certainly generate some data. Fortunately, Steam allows access to said data by providing various Application Programming Interfaces (APIs), allowing publishers and researchers alike to generate their own sets of data and gain meaningful insights [6]. As a result, a lot of studies on the platform have appeared over the past few years from various disciplines, including consumer behavior [7], human-computer interaction [8], economics [9], education [10], health [11], social sciences [12], law [13], business [14], and gaming specific engagement [15].

While publishers have access to proprietary APIs that provide sensitive, non-public user data, general APIs still allow access to publicly available data on platforms, including game prices, genres, reviews (game-specific data), as well as player activity, game ownership, and achievement statistics (public user data) [16]. This can be useful for business applications through data-driven analysis and can also be used in machine learning algorithms to predict various variables, such as game prices, discount trends, player count, game ratings, and many others.

Predicting daily player count is essential for game developers and publishers, as it provides insights into player activity, engagement, and behavior, enabling informed decision-making. Therefore, it is important for the prediction model to be as accurate as possible to avoid any potential misinformation. The features used play an important role in the model's predictive performance and must provide meaningful information for the model. Existing studies on predicting daily player count used various metadata features in addition to historical data, such as game's genres, supported languages, number of achievements, etc. However, those features stay the same most of the time, and don't give enough temporal context for the model, as historical data like daily player count can have patterns, such as trends and seasonality depending on the time of the observation. But the historical data can be transformed into lagged features using feature engineering to capture such historical patterns and enhance the prediction performance of the model. However, there is a limited number of studies in exploring the use of time-lagged features for predicting player count.

In this paper, we proposed a new method utilizing lagged features on top of existing metadata features to accurately predict daily player count on Steam. Historical player count data were transformed into new lagged features using the sliding window technique, providing more temporal context than using meta-data only. This will be explained further in Section III. Several combinations of feature selection methods and machine learning models were tested and compared on their predictive performance to find out which one gives the most accurate prediction. The models are Random Forest, Support Vector Regression (SVR), and XGBoost, with feature

selection methods such as Pearson's Correlation feature selection, Recursive Feature Elimination (RFE), and the models' embedded feature selection method. The structure of the paper is as follows: Section I presents the background of this study, Section II discusses the works related to this study, Section III explains the methodology, Section IV the results and discussion, and Section V contains the conclusion of this study and things to be addressed for future works.

II. Related Works

As stated in Section I, there have been a lot of studies on the Steam platform in the past few years. A study by Prathama et al. [6] created a system to provide data analysis on current game trends and predict game trends for the next two weeks using game and user data obtained with Steam API. The trend prediction is through predicting future game rating and future player count, using Multiple Linear Regression (MLR) method. H. Zhang [16] used the same MLR method to predict game sales, and also investigated various factors and how they are related with game sales. Zendle et al. [7] analyzed trends on ingame microtransactions using historical data from 463 most-played games in the Steam platform.

Wannigamage et al. [17] analyzed the changes in player population and weekly player count patterns during the COVID-19 pandemic, and also analyzed the changes in game sales. They also tried to identify which games that became popular during the pandemic by comparing player population from before and during the pandemic. Vuorre et al. [2] also analyzed the changes in players behavior during COVID-19 pandemic using various data from popular games, like play time and player count from both before the pandemic and during the pandemic. Wu et al. [18] conducted an analysis on the impact of the COVID-19 pandemic on the video game industry overall, by analyzing and comparing the number of games released and also player count from before the pandemic and during the pandemic, and also predicted the demand for online games with machine learning, using historical player data combined with COVID-19 features and human mobility features to predict daily player count. Several machine learning models were used, including SVR, Random Forest, and Ridge Regression.

Varghese et al. [19] discuss an online game's success upon release using the game's historical player data on the Steam platform with models including SVR, Random Forest, and Bayesian Regression. Teja et al. [20] compared various machine learning algorithms and predicted the rating from Metacritic for games on Steam by comparing variables that are related to the score, like genres and player count. Abdul-Rahman et al. [21] developed a model for churn prediction using Vector Autoregression enhanced with sentiment analysis on user reviews from various games on the Steam platform.

III. Methodology

This section explains the data and methods used in the study. The workflow is depicted in Fig. 1. The data was gathered from various public sources. The gathered data was then pre-processed to get it ready for machine learning models. New lagged features were created using the sliding window technique, and several feature selection methods were selected to select the most relevant features: Pearson's Correlation feature selection, RFE, and embedded method. The selected features from each method were used in three machine learning models: Random Forest, SVR, and XGBoost. Grid Search hyperparameter tuning is used to find the best parameters for each model. The results from each combination were then compared to find out which one is the best in predictive performance.

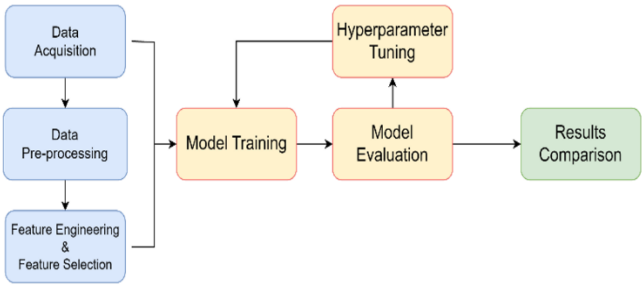

Fig. 1. Workflow of the study.

A. Data Acquisition

This study uses publicly available data from various sources. The historical dataset was collected through manual download from SteamDB, a third-party database providing information on games from the Steam platform. The historical dataset consisted of player count, number of positive and negative reviews from 100 most-played games on Steam, sorted by peak number of players during the date of collection, 8 August 2024. All observations were in UTC time zone. Then a code was made for scraping metadata from each game. The code was written in python programming language. It works by sending an HTTP request using the Steam 'getdetails' API to get each game's details, utilizing the unique ID number from each game called appid. The details to be extracted were pre-determined to avoid getting unnecessary information. This process was repeated for all games. Then all the data from the games was parsed and then compiled into a single CSV file. The metadata included genres, release date, supported languages, and many more. Over 20 columns of raw data were collected using the API.

B. Data Pre-processing

Since the raw data were collected from different sources, it needed to be pre-processed to be ready for model training. Missing values on the metadata were removed, and several features were transformed into new ones with feature engineering. Table I shows an example of new features created from the raw metadata. For historical data, three new features were created. First was the game's age since the release date, then Cumulative Moving Average (CMA) for both positive and negative reviews. All the new features were calculated for each observation. The result was a total of six features for the historical data excluding the time variable. Table II shows an example of the resulting data from the game Counter-Strike 2, one of the games on the dataset.

Additional features were created for the historical data using the sliding window technique. The sliding window technique transforms the time-series into a supervised learning problem by shifting the data, taking prior observations as

lagged features, depending on the window size. For example, the current observation is labeled as *x* and the sliding windows technique was applied to capture observations for the past 30 days, the data would be shifted 30 days, and the window would contain observations from *x*-29 to *x*, with *x* being the most recent. In this research, a window size of 7 was randomly chosen, meaning that observations from previous seven days were used as lagged features. The technique was applied to all features on the historical data, resulting in a total of 42 features. Then the data was shifted once more to obtain the target feature *x*+1, which was the observation on the next day. Then the processed historical data was merged with the metadata, and then split into training/validation/testing sets, with a ratio of 60/20/20. The split was done to ensure the model robustness against unseen data. All features were normalized after split using Min-Max scaler, including the target feature.

TABLE I. EXAMPLE OF NEW FEATURES CREATED FROM METADATA

name	app_id	required_age	is_free	dlc	achievements	full_controller_support
Grand Theft Auto V	271590	17	0	1	77	1
No Man's Sky	275850	0	0	0	27	1
BeamNG.drive	284160	0	0	0	4	0
Sid Meier's Civilization® VI	289070	0	0	10	320	0

TABLE II. EXAMPLE OF HISTORICAL DATA FROM COUNTER-STRIKE 2

DateTime	Players	Positive reviews	Negative reviews	Positive_CMA	Negative_CMA	days_since_release
2024-03-14	1447897	2414	-862	2025.5721809169765	-304.2506195786865	4223
2024-03-15	1474990	0	0	2024.9448745741715	-304.1563951687829	4224
2024-03-16	1490175	2468	-858	2025.0820433436531	-304.32786377708976	4225
2024-03-17	1425033	0	0	2024.4552770040236	-304.23367378520584	4226

C. Feature Selection

This study uses three feature selection methods, which are Pearson's Correlation Feature Selection for filter method, Recursive Feature Elimination (RFE) for wrapper method, and embedded feature selection method from the model itself.

1) Pearson's correlation feature selection: This feature selection method uses Pearson's Correlation Coefficient (PCC), which measures linear relationship between two or more variables. The correlation value *r* between variables *X* and *y* can be obtained through Eq. (1).

$$r = \frac{\Sigma_{i=1}^{n}(X_i-\bar{X})(y_i-\bar{y})}{\sqrt{\Sigma_{i=1}^{n}(X_i-\bar{X})^2}\cdot\sqrt{\Sigma_{i=1}^{n}(y_i-\bar{y})^2}} \quad (1)$$

where $\bar{X}$ and $\bar{y}$ are mean values of variables *X* and *y*. The correlation value ranges from -1 to +1, where values closer to -1 or +1 indicate stronger correlation, and values closer to 0 indicate weaker correlation. This method selects features with correlation value higher than a certain threshold.

2) Recursive Feature Elimination (RFE): This feature selection method selects the most important features in the dataset by recursively removing the least important features until the number of features to select is reached.

3) Embedded method: Embedded method refers to feature selection method that is built-in to the model itself. The model performs feature selection during training. Models based on decision tree like Random Forest and XGBoost use feature importance to select the most relevant features.

D. Machine Learning Models

This study used three machine learning models: Random Forest (RF), Support Vector Regression (SVR), and XGBoost.

1) Random forest: Random Forest is an ensemble model of decision trees that use random sub-samples from a dataset for prediction, and can be used for both classification and regression tasks [19]. For regression, the model takes predictions from all decision trees then averages them for the final result.

2) Support Vector Regression (SVR): Support Vector Regression (SVR) is a type of Support Vector Machine (SVM) that is used for regression tasks, and is a commonly used method for time-series forecasting [22]. It tries to fit an optimal hyperplane for predicting continues values.

3) XGBoost: XGBoost stands for Extreme Gradient Boosting. It can also be used in both classification and regression tasks. XGBoost is based on Gradient Boosting algorithm, where the decision trees are added to the model sequentially.

E. Model Evaluation

This study used and compared the combinations of feature selection methods machine learning models. Three machine learning models were used and tested: Random Forest (RF), Support Vector Regression (SVR), and XGBoost. The models were first trained using default parameters, then tuned using Grid Search hyperparameter tuning to find the best settings for each model. We used four evaluation metrics to evaluate the model's performance: Coefficient of Determination (R^2), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE),

and Mean Absolute Percentage Error (MAPE), all of which can be defined in Eq. (2), (3), (4), and (5) respectively.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (2)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (3)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \quad (4)$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \quad (5)$$

where, y_i is the actual value, $\hat{y}_i$ is the predicted value, and $\bar{y}_i$ is the mean value.

IV. Results and Discussion

A. Training and Validation Results

The results for model training with default parameters along with the validation can be seen on Table III, Table IV, and Table V. The results were grouped based on the feature selection method for readability purposes. The best results from each method are highlighted in bold.

TABLE III. Training and Validation – Pearson's Feature Selection

Model	Training				Validation			
	R²	***RMSE***	***MAE***	***MAPE***	***R²***	***RMSE***	***MAE***	***MAPE***
RF	**0.9971**	**22568.1233**	**10925.2073**	**5.81%**	0.9932	25517.8834	13777.1053	6.70%
SVR	0.9913	13470.8283	5308.5905	6.05%	0.8868	6192.4275	3751.0109	7.01%
XGBoost	0.9959	26876.6614	13251.0304	8.11%	**0.9943**	**23282.8510**	**12679.5643**	**6.69%**

TABLE IV. Training and validation – RFE

Model	Training				Validation			
	R²	***RMSE***	***MAE***	***MAPE***	***R²***	***RMSE***	***MAE***	***MAPE***
RF	**0.9983**	**17410.7001**	**9081.5890**	**4.89%**	**0.9938**	**24321.2331**	**13347.6318**	19.29%
SVR	0.9941	32088.4842	19245.7056	31.47%	0.9930	25748.6940	17236.5587	22.32%
XGBoost	0.9969	23422.6130	13053.8578	8.72%	0.9936	24719.3096	14251.4393	**10.08%**

TABLE V. Training and validation – Embedded Feature Selection

Model	Training				Validation			
	R²	***RMSE***	***MAE***	***MAPE***	***R²***	***RMSE***	***MAE***	***MAPE***
RF	**0.9962**	**25747.2753**	12930.4714	6.93%	0.9913	28877.6052	15217.6164	6.98%
XGBoost	0.9962	26001.9723	**11681.7900**	**6.20%**	**0.9945**	**22880.2620**	**11970.0420**	**5.91%**

B. Testing Results

The models were then put to test to see how well the model generalizes with truly unseen data using the testing set. The results of the experiments can be seen on Table VI, Table VII, and Table VIII. The results were grouped based on the feature selection method for readability purposes. Models with all the features were also tested for comparison and can be seen on Table IX. The best results from each method are highlighted in bold.

TABLE VI. Experiment Results – Pearson's Feature Selection

Model	R^2	RMSE	MAE	MAPE
RF	**0.9943**	**30926.7590**	**16087.1698**	**5.49%**
SVR	0.8280	60648.5777	21614.1249	10.21%
XGBoost	0.9925	35243.8582	18644.4787	6.72%

TABLE VII. Experiment Results – RFE

Model	R^2	RMSE	MAE	MAPE
RF	0.9933	33282.8687	18839.3967	19.90%
SVR	**0.9937**	**32290.6700**	**20560.2261**	14.30%
XGBoost	0.9845	50718.4633	25457.2603	**11.26%**

TABLE VIII. Experiment Results – Embedded Feature Selection

Model	R^2	RMSE	MAE	MAPE
RF	0.9900	40899.9580	22687.4373	8.09%
XGBoost	**0.9926**	**35068.2627**	**17777.9998**	**5.90%**

TABLE IX. Experiment Results – No Feature Selection

Model	R^2	RMSE	MAE	MAPE
RF	**0.9934**	**33140.1820**	**18877.3783**	**17.72%**
SVR	0.9926	34990.9246	22427.8018	23.58%
XGBoost	0.9766	62394.0293	32761.8084	23.78%

Random Forest combined with Pearson's Correlation for feature selection gives the best results overall, with an R^2 score of 0.9943, a slight improvement from the model using all features with an R^2 score of 0.9934. Fig. 2 shows the prediction error of the model. This indicates that even without feature selection, the model was still able to explain more than 95% of the variance. The MAPE also dropped significantly from 17.72% to 5.49%. XGBoost seemed to benefit from feature selection the most, based on the improved results compared to no feature selection.

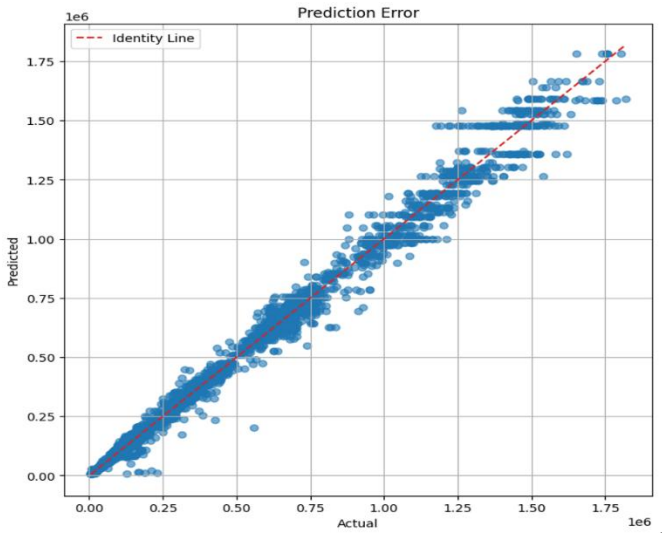

Fig. 2. Prediction error for Random Forest – Pearsons' Feature Selection

C. Results Comparison

A previous study using a linear model [16] had achieved an R^2 score of 0.6756, indicating that the model managed to capture 67.56% of the variation. The results indicated that linear models might not be suitable for predicting the number of players. Another study [18] tested various models, including SVR, RF, and Ridge Regression, with Pearson's Correlation feature selection method. The method reduced the number of features to be used in the models from 889 to 163. The result achieved with the best model was an R^2 score of 0.805, indicating that the model managed to capture 80.5% of the variation. The best model in this study achieved an R^2 score of 0.9943, and an average R^2 score above 0.9 across all models, a better result compared to the previous studies mentioned above.

V. Conclusion and Future Work

In this paper, we presented a method using lagged features in predicting daily player count, using historical data from video games on the Steam platform. The lagged features were created using sliding window method, using data from the last 7 days. Several feature selection methods and machine learning models were tested. Random Forest and Pearson's Correlation Feature Selection shows the best predictive performance amongst all combinations. Despite that, all the other combinations have an average R^2 score above 0.9, showing the effectiveness of our method.

However, there are several limitations in this study. The data used was from video games with the most player populations at the time of the study, so it might be different in the future. This study didn't consider the price of each video game, but only whether they were free or not. Future works may consider adding individual game prices, and the price during sales for more detailed analysis. Game ratings obtained from sentiment analysis of user reviews may also be used for more accurate results.

References

[1] Y. S. Balhara, D. Kattula, S. Singh, S. Chukkali, and R. Bhargava, "Impact of lockdown following COVID-19 on the gaming behavior of college students," Indian J Public Health, vol. 64, no. 6, p. 172, 2020, doi: 10.4103/ijph.IJPH_465_20.

[2] M. Vuorre, D. Zendle, E. Petrovskaya, N. Ballou, and A. K. Przybylski, "A Large-Scale Study of Changes to the Quantity, Quality, and Distribution of Video Game Play During a Global Health Pandemic," Technology, Mind, and Behavior, vol. 2, no. 4, 2021, doi: 10.1037/tmb0000048.

[3] E. Haug et al., "Increased Gaming During COVID-19 Predicts Physical Inactivity Among Youth in Norway—A Two-Wave Longitudinal Cohort Study," Front Public Health, vol. 10, Feb. 2022, doi: 10.3389/fpubh.2022.812932.

[4] M. N. Rizani, M. N. A. Khalid, and H. Iida, "Application of Meta-Gaming Concept to the Publishing Platform: Analysis of the Steam Games Platform," Information, vol. 14, no. 2, p. 110, Feb. 2023, doi: 10.3390/info14020110.

[5] T. Guzsvinecz and J. Szűcs, "Length and sentiment analysis of reviews about top-level video game genres on the steam platform," Comput Human Behav, vol. 149, p. 107955, Dec. 2023, doi: 10.1016/J.CHB.2023.107955.

[6] N. Y. Prathama, R. Asmara, and A. R. Barakbah, "Game Data Analytics using Descriptive and Predictive Mining," in 2020 International Electronics Symposium (IES), IEEE, Sep. 2020, pp. 398–405. doi: 10.1109/IES50839.2020.9231949.

[7] D. Zendle, R. Meyer, and N. Ballou, "The changing face of desktop video game monetisation: An exploration of exposure to loot boxes, pay to win, and cosmetic microtransactions in the most-played Steam games of 2010-2019," PLoS One, vol. 15, no. 5, 2020, doi: 10.1371/journal.pone.0232780.

[8] C. Phillips, M. Klarkowski, J. Frommel, C. Gutwin, and R. L. Mandryk, "Identifying Commercial Games with Therapeutic Potential through a Content Analysis of Steam Reviews," Proc ACM Hum Comput Interact, vol. 5, no. CHI PLAY, pp. 1–21, Oct. 2021, doi: 10.1145/3474682.

[9] A. M. Thorhauge and R. K. L. Nielsen, "Epic, Steam, and the role of skin-betting in game (platform) economies," Journal of Consumer Culture, vol. 21, no. 1, pp. 52–67, Feb. 2021, doi: 10.1177/1469540521993929.

[10] C. Moro, C. Phelps, and J. Birt, "Improving serious games by crowdsourcing feedback from the STEAM online gaming community," Internet High Educ, vol. 55, p. 100874, Oct. 2022, doi: 10.1016/J.IHEDUC.2022.100874.

[11] A. O. Thunström, I. Sarajlic Vukovic, L. Ali, T. Larson, and S. Steingrimsson, "Prevalence of virtual reality (VR) games found through mental health categories on STEAM: a first look at VR on commercial platforms as tools for therapy," Nord J Psychiatry, vol. 76, no. 6, pp. 474–485, Aug. 2022, doi: 10.1080/08039488.2021.2003859.

[12] J. Kohlburn, H. Cho, and H. Moore, "Players' perceptions of sexuality and gender-inclusive video games a pragmatic content analysis of steam reviews," Convergence: The International Journal of Research into New Media Technologies, vol. 29, no. 2, pp. 379–399, Apr. 2023, doi: 10.1177/13548565221137481.

[13] L. Y. Xiao and L. L. Henderson, "Illegal video game loot boxes with transferable content on steam: a longitudinal study on their presence and non-compliance with and non-enforcement of gambling law," Int Gambl Stud, pp. 1–27, Aug. 2024, doi: 10.1080/14459795.2024.2390827.

[14] A. M. Thorhauge, "The steam platform economy: From retail to player-driven economies," New Media Soc, vol. 26, no. 4, pp. 1963–1983, Apr. 2024, doi: 10.1177/14614448221081401.

[15] K. Stecuła, "Analysis of asymmetric VR games – Steam platform case study," Technol Soc, vol. 78, p. 102673, Sep. 2024, doi: 10.1016/J.TECHSOC.2024.102673.

[16] H. Zhang, "The Establishment of Multi-variable Linear Regression in Steam Sales," in Proceedings of the 2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022), 2022, pp. 853–856. doi: 10.2991/aebmr.k.220307.137.

[17] D. Wannigamage, M. Barlow, E. Lakshika, and K. Kasmarik, "Analysis and Prediction of Player Population Changes in Digital Games During the COVID-19 Pandemic," 2020, pp. 458–469. doi: 10.1007/978-3-030-64984-5_36.

[18] S. Wu, H. Hu, Y. Zheng, Q. Zhen, S. Zhang, and C. Zhan, "The Impact of COVID-19 on Online Games: Machine Learning and Difference-in-Difference," Communications in Computer and Information Science, vol. 1492 CCIS, pp. 458–470, 2022, doi: 10.1007/978-981-19-4549-6_35.

[19] R. R. Varghese, D. R. Aiswarya, A. Roy, V. Muraly, and S. Renjith, A Novel Approach to Predict Success of Online Games Using Random Forest Regressor for Time Series Data, vol. 881. 2022. doi: 10.1007/978-981-19-1111-8_3.

[20] A. S. Teja, M. L. I. Hanafi, and N. N. Qomariyah, "Predicting Steam Games Rating with Regression," E3S Web of Conferences, vol. 388, p. 02001, May 2023, doi: 10.1051/e3sconf/202338802001.

[21] S. Abdul-Rahman, M. F. A. M. Ali, A. A. Bakar, and S. Mutalib, "Enhancing churn forecasting with sentiment analysis of steam reviews," Soc Netw Anal Min, vol. 14, no. 1, 2024, doi: 10.1007/s13278-024-01337-3.

[22] J. M. Valente and S. Maldonado, "SVR-FFS: A novel forward feature selection approach for high-frequency time series forecasting using support vector regression," Expert Syst Appl, vol. 160, p. 113729, Dec. 2020, doi: 10.1016/j.eswa.2020.113729.