IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 September 2024, accepted 3 October 2024, date of publication 10 October 2024, date of current version 21 October 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3477624

RESEARCH ARTICLE

Predictive Analytics of In-Game Transactions: Tokenized Player History and Self-Attention Techniques

MILOŠ A. KOVAČEVIĆ[1], MARKO D. PEŠOVIĆ[1], ZORAN Z. PETROVIĆ[2], AND ZORAN S. PUCANOVIĆ[1]
[1]Faculty of Civil Engineering, University of Belgrade, 11000 Belgrade, Serbia
[2]Faculty of Mathematics, University of Belgrade, 11000 Belgrade, Serbia

Corresponding author: Marko D. Pešović (mpesovic@grf.bg.ac.rs)

This work was supported by the Linear Algebra and Artificial Intelligence, Algebra AI, Belgrade, Serbia.

ABSTRACT Players' purchases in free-to-play online games often serve as crucial indicators of user engagement and behavior. Understanding these purchases not only enhances the personalization of the gaming experience but also enables the optimization of game monetization strategies. This paper introduces a methodology for predicting players' purchases using Transformers neural networks based on the Self-Attention technique, customized for processing sequential data. By discretizing the values of features representing a player's history and leveraging tokenized inputs related to the discretized history, the methodology aims to forecast whether a player will make a purchase within the next 3, 5, or 7 days. The proposed approach is further validated by comparing its performance with commonly adopted machine learning techniques such as Random Forest, XGBoost, and Multilayer Perceptron, demonstrating its advantages in predicting player purchases.

INDEX TERMS In-game purchases, prediction, self-attention, transformers.

I. INTRODUCTION

In the contemporary digital world, scrutinizing user transactions holds most important significance across different sectors, spanning from banking and e-commerce to financial services and online games. The ability to predict whether a user will undertake a particular transaction has escalated in importance, serving as a linchpin for refining the user experience, mitigating fraud, and enhancing operational efficiency within businesses.

In a free-to-play online game (F2P game), which represents the type of non-contractual business model, players can download, install, and play for free. Still, they can make in-app purchases, such as virtual currency or temporary boosts, or remove advertisements. Revenue in free-to-play games mainly stems from commercials and in-game purchases, with more permanent user engagement directly correlating to higher revenue likelihood.

In online gaming, a small fraction of users drives the majority of sales, while non-paying users primarily consist of those who discontinue playing [1]. Consequently, predicting whether a player will make purchases in the near future is essential for shaping effective business policies within the gaming industry. Developing predictive models for player purchases enables game developers to implement personalized marketing strategies, efficiently optimize resource allocation, and enhance player retention efforts. Such a proactive approach not only maximizes revenue but also significantly boosts player satisfaction, thereby creating a competitive advantage in the dynamic landscape of gaming.

This research aims to unravel the intricacies of player behavior within an F2P game, identifying patterns in a player's history that indicate potential purchases. It uses "Woka Woka", developed and published by the gaming company "Two Desperados" (https://twodesperados.com),

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Bellan.

as an example of an F2P game where players can purchase virtual currency for rewinds and booster options. The research primarily contributes by introducing a novel discretization method for players' histories and applying Transformers in this context, paving the way for innovative approaches to in-game monetization strategies.

In contemporary data analysis of games, particularly in F2P and mobile games, several key studies offer valuable insights into player behavior and decision-making. For example, [2] and [3] investigate playing patterns using time series clustering to identify similarities among players. These methods enable personalized player retention strategies and provide insights into different player groups. Specifically, [2] utilizes K-Means and Hierarchical Agglomerative Clustering with Dynamic Time Warping for measuring similarity between time series. On the other hand, [3] applies K-Means and DBSCAN clustering methods, coupled with Principal Component Analysis for dimensionality reduction. Additionally, [4] explores using self-organizing maps to develop player models based on in-game actions in "Tomb Raider: Underworld", aiming to automatically generate player profiles and better tailor the game to diverse player types.

A comprehensive grasp of online customer behavior and its prediction through deep learning methods on extensive data, as detailed in [5], [6], [7], and [8], formed the basis for exploring purchases within online games. The paper [9] presents a Lexicon-Based Analysis for predicting the currency market in the "World of Warcraft" game. They proposed a method for predicting the next-day rise and a method for predicting daily fluctuations in the price of a currency.

In [10], the authors focus on forecasting player retention, seeking to reveal distinctive strategies for companies to amplify revenue and incentivize players to either persist or embark on a new "life cycle" within another game. They implemented a Hidden Markov Model [11] to address temporal dynamics and found that a Neural Network achieved the best prediction performance. In [12], the authors explored how to predict when players would disengage and when they would make their first purchase using event frequency data. They found that it is sufficient for prediction to use only the frequency of events, without utilizing any other information about them. When predicting initial purchases, they observed player behavior in the preceding two weeks, assuming the player had not made any prior in-game purchases. Article [13] explores forecasting purchasing behaviors within F2P mobile games. The authors investigate player histories, with the tracking period initiating at random points within the first seven days of installation and lasting 30 days. The primary objectives were to discern whether players would opt to pay for in-game content (a classification task), thus facilitating the initial categorization of players. They recommended Decision Trees, Random Forests, and Support Vector Machines [14]. In [15], authors used the same machine learning techniques for predicting purchases in a game. Additionally, study like [16] use Logistic Regression, Random Forest, and Gradient Boosting models to predict purchases in F2P games based on playing frequency, previous purchases, and types of interactions in the game. They also use Long Short-Term Memory Networks to analyze sequences and player behavior over time. Like Recurrent Neural Networks, they are effective for time series predictions due to their ability to capture short-term dependencies.

In [17], the authors apply Logistic Regression and Gradient Boosting methods to predict player churn under real-world conditions, focusing on features such as playtime, session length, and session intervals. Similarly, [18] employs Random Forest, Gradient Boosting, and Deep Learning techniques to identify high-risk player groups and recommend strategies for companies to enhance player retention.

The Decision Tree is the most commonly recognized method for customer churn prediction, see [19]. In [20] and [21], the application of the Random Forest technique [22] for churn prediction demonstrated its superiority over alternative machine learning methods. Random Forests, as ensembles of decision trees, are frequently selected for churn prediction due to their high generalization ability, ability to identify informative features, and capacity to handle missing values, correlations, and high-dimensional data, see [23], [24].

Besides Random Forest, in contemporary predictive modeling, a special type of Neural Networks, Multilayer Perceptrons, XGBoost, and Support Vector Machines are commonly used due to their distinct benefits. Multilayer Perceptrons excel at capturing complex, non-linear relationships and are effective with high-dimensional data. XGBoost, a powerful gradient boosting algorithm, provides high accuracy and efficient training, particularly for imbalanced datasets and complex feature interactions. Support Vector Machines, as maximum margin classifiers, are capable of finding the optimal class boundaries.

However, all of these techniques, except Long Short-Term Memory Networks (LSTM) and Recurrent Neural Networks (RNN), require fixed-length input vectors thus complicating the modeling of variable-length player histories. While LSTMs and RNNs are well-suited for handling variable-length inputs, they are less effective at modeling long-term dependencies due to the vanishing and exploding gradient problems encountered during training (sequences greater than 300 timesteps). Support Vector Machines face memory and execution speed limitations when trained on large datasets, as they require the storage of a precomputed distance matrix between all pairs of input data, resulting in a memory complexity of $O(n^2)$.

This paper assumes that a recent history significantly influences future players' behavior. However, similar recent patterns could be interpreted differently, depending on the patterns in the whole player's history (see Fig. 1).

Therefore, it is necessary to design a model that overcomes the limitations of all previously mentioned techniques by effectively handling variable-length player histories as inputs and detecting long-span relationships between behavioral patterns.

One such model, used in this research, is the Transformer Neural Network [25] that utilizes self-attention among input data [26], [27], [28]. Transformer Neural Networks (TNN) have proven highly effective in processing sequential data, such as texts or time series, and thus represent a promising approach for analyzing player purchases. TNNs facilitate input vectors of varying lengths, effortlessly establishing long-span relationships within the input data, and do not require feature selection since it is implicitly performed in hidden layers of the network.

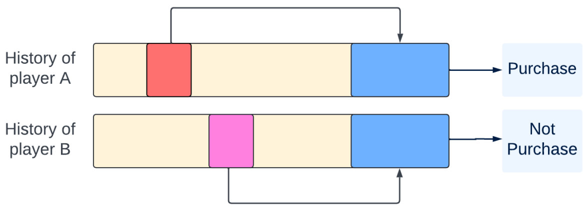

FIGURE 1. The same recent history does not always imply the same purchase decision, as different behaviors from the past could significantly influence future player actions. Different colors represent different behaviors.

TNNs are used in [8] to predict the next purchase day for e-commerce customers, and [29] for creating a churn prediction model. In addition, authors in [8] compare TNN performance with traditional methods such as ARIMA [30], XGBoost [31], and LSTM [32].

This paper explores various aspects of applying the TNN model in predicting player purchases in F2P games. Section II-A briefly describes the "Woka Woka" dataset obtained from the gaming company "Two Desperados". Section II-B. discusses three representations of players' history to predict whether a player will purchase in the upcoming period. The representations were crafted for flexibility and suitable for application in four models tested in this paper: Random Forest (RF), XGBoost (XGB), Multilayer Perceptron (MLP), and TNN. Section II-C. demonstrates the architecture of the TNN model used in this study. Section III provides a detailed description of the experiment, explaining the methods and measures used to evaluate and compare the obtained results of proposed classification models and illustrate the advantages of the TNN model. Section IV contains a conclusion of the experiments and suggestions for further research on a similar topic.

II. PREDICTING PURCHASES FOR PLAYERS

A. DATASET DESCRIPTION

The data were collected from the mobile game "Woka Woka". The dataset was created by selecting users who registered between October 1st, 2022, and April 30th, 2023, and their actions were observed over the first 90 days.[1]

The data obtained for each player included detailed information about their daily behavior. In addition to active/inactive status and purchase/non-purchase activity on a given day, the data encompassed the number of sessions, total time spent playing, and the number of levels either played or completed that day. The dataset further distinguished between levels played for the first time and those revisited or previously completed. Additionally, it provided insights into the amount of in-game currency purchased, earned, and spent during the day, as well as the player's remaining stash at the end of the day.

The database contained 977 287 registered players and tracked their characteristics over 90 days from the registration date. Out of that number, only 13 765 (1.4%) had made at least one transaction. Table 1 presents the basic statistics on the number of active days for all players and players who made at least one purchase.

The reliability of any purchase prediction model depends on the quality of the data, so feature selection is a crucial stage in developing a model. Players who did not show payment activity within the game were excluded from the dataset. Keeping registered players without any payment activity would skew various variables and produce samples lacking historical observations for analysis. Therefore, the model presented in this paper tracks and predicts player purchases from their first purchase until seven days after their last puchase - beneficial players from the company's point of view. The total number of beneficial players who made at least one purchase during the specified period was 13 765, while the total number of purchases in that period amounted to 86 785. Table 2 displays basic statistics of purchase counts per beneficial player and per day.

After excluding players with no purchases, each day in a beneficial player's history is represented with the features described in Table 3 (in further text, a beneficial player is denoted as a 'player').

TABLE 1. Basic statistics on the number of active days for all players and players who made at least one purchase. Players who made at least one purchase are obviously more active than others.

Statistics	All players	Players who made a purchase
min	1	1
Q1	1	13
median	2	35
Q3	6	68
max	90	90
mean	8.4660	40.4940
std	16.6176	29.4043

The process of selecting informative features for the purchase prediction task focused on three groups of predictors: player engagement, player skill, and willingness to make

[1]This dataset and the code for the proposed Transformer model could be downloaded from: https://github.com/mpesovicgrfbgacrs/TransformerInGamePurchases

TABLE 2. **Basic statistics of the purchase counts per beneficial player (13 765) and per day (90).**

No. Purchases	Per player	Per day
min	1	521
Q1	1	680
median	3	910
Q3	7	1187
max	83	1780
mean	6.3048	953.6813
std	953.6813	316.6922

in-game purchases. Data regarding inter-player interaction was absent, as the game does not support such features. Specifically, features F_1, F_3, and F_4 reflect the player's level of engagement in the game. Features F_5, F_6, F_7, along with F_{12} and F_{13}, provide insights into various aspects of player skill. Meanwhile, features F_2 and F_8 through F_{11} evaluate the player's overall commitment to the game and their behavioral patterns related to potential monetization.

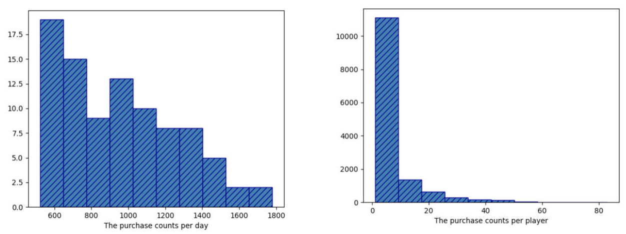

FIGURE 2. **The histogram shows the number of purchases per day and per beneficial player.**

TABLE 3. **Description of the features describing a player's behavior in each day of his/her history.**

Feature label	Description of the feature
F_1	the indication of whether there has been any activity on the day or not (0/1)
F_2	the indication of whether there has been any purchase on the day or not (0/1)
F_3	total time spent in the game since the registration date
F_4	total number of sessions on the day
F_5	total number of levels passed since the registration
F_6	percentage of different levels passed compared to the total number of different levels played on the day
F_7	percentage of replayed levels compared to the total number of played levels on the day
F_8	gold (in-game currency) purchased by players on the day
F_9	gold (in-game currency) gained earned as a reward on the day
F_{10}	gold (in-game currency) spent on the day
F_{11}	gold (in-game currency) in stash on the day
F_{12}	rewinds used on the day
F_{13}	boosters used on the day

B. PROPOSED REPRESENTATIONS OF A PLAYER'S HISTORY

Using the features of the player p from Table 2, player's history is represented by a real matrix H^p. The rows of the matrix H^p represent features and the columns denote consecutive days in the player's history from registration day $t = 1$ to the day $t = T^p$, where T^p denotes the last day in p's history. So, H^p is the $13 \times T^p$ matrix determined by

$$H^p = [h_{i,t}]_{13\times T^p},$$

where $h_{i,t}$ is the numerical value of the feature F_i from Table 2 on day t, $1 \leq t \leq T^p$. Note that different players have different values for T^p (variable length histories). The aim is to predict whether a purchase will happen within the next k days (3, 5, or 7 in this case study). If y_t^p is the value of the binary feature F_2 (Table 2) for the player p on day t, then the main goal is to find a classification function such that:

$$H^p \mapsto y^p \in \{0, 1\}, \quad y^p = \max\{y^p_{T^p+1}, y^p_{T^p+2}, \ldots, y^p_{T^p+k}\},$$

where $y^p = 1$ indicates that a player p will purchase in the next k days.

To accommodate H^p to be fed as an input to the RF-based, XGB-based, MLP-based, and TNN-based classification models, the matrix H^p is tranformed into a vector of continuous values (Feature Statistics Representation, Token Frequency Representation) and a sequence of discrete values (Feature Tokens Representation).

1) FEATURE STATISTICS REPRESENTATION

This representation transforms H^p into a vector

$$x^p = (T^p, \mathrm{freq}^p, \mathrm{stat}^p_2, \ldots, \mathrm{stat}^p_i, \ldots, \mathrm{stat}^p_13),$$

where T^p represents the duration of the observed player history, freq^p represents the mean of the first row of H^p (the percentage of days when the player p played the game), and for $2 \leq i \leq 13$ the vector stat^p_i is represented by

$$(\min(H^p_{i,\rightarrow}), \mathrm{Q1}(H^p_{i,\rightarrow}), \mathrm{Q}_2(H^p_{i,\rightarrow}), \mathrm{Q3}(H^p_{i,\rightarrow}), \max(H^p_{i,\rightarrow})),$$

where $H^p_{i,\rightarrow}$ are the values of the i-th row of H^p (i.e. F_i values of p's history). Note that the vector x^p has 62 coordinates.

2) FEATURE TOKENS REPRESENTATION

TNNs are developed as natural language models whose basic input units are categorical values - tokens (pieces of words, or whole words). To use this model the input data from H^p must be converted into categorical values. This can be done by discretizing the feature values in H^p.

Each player's day (a column in H^p) is represented by a sequence of 13 tokens, where each token reflects the values of the corresponding feature from Table 2 on that day. The first two binary features, F_1 and F_2 are represented by one of two tokens, indicating whether the corresponding feature is 0 or 1. The remaining features F_i, for $2 < i \leq 13$, are represented by one of four tokens depending on the feautre's value:

1) Token1 - represents the absence of F_i's value due to player inactivity during the day,
2) Token2 - represents that F_i's value is an outlier,
3) Token3 and Token4 - represent that the value of F_i belongs to one of the two characteristic ranges inside the domain of feature F_i.

The characteristic ranges for each feature are independently found using the K-Means algorithm [33] with $K = 2$. The algorithm is applied to the actual feature values of all players across all days with player activity (i.e., the i-th rows of all history matrices). Given that the K-Means algorithm is sensitive to outliers, the Interquartile Range Method [34] is applied separately for each feature F_i to detect and mitigate extreme values before clustering. For multidimensional data, K-Means typically requires normalization or standardization for effective clustering. However, this preprocessing is unnecessary for one-dimensional data.

Instead of the original matrix H^p, the corresponding tokenized history matrix TH^p will be created by tokenizing values from H^p as described above. The historical representation of each player p is the sequence x^p that is arisses obtained by concatenating the columns of the matrix TH^p. Namely, sequence x^p has a form

$$x^p = (TH^p_{1,\downarrow}, \ldots, TH^p_{t,\downarrow}, \ldots, TH^p_{T^p,\downarrow}),$$

where $TH^p_{t,\downarrow}$ is the column of TH^p containing tokens representing the day t in p's history. Note that x^p is a sequence of length $13T^p$ containing no more than 48 tokens that form a token vocabulary.

3) TOKEN FREQUENCIES REPRESENTATION

To use tokenized data from the matrix TH^p for RF, XGB, and MLP models, it is necessary to reduce it to a vector of constant length. This is accomplished in a two-step process: firstly, a Feature Token's representation sequence is constructed as described in the previous section; finally, this sequence is converted into a frequency vector x^p in which the i-th coordinate represents the normalized frequency of the i-th token from the token vocabulary. Note that the resulting x^p has 48 coordinates (48 distinct tokens in the token vocabulary).

C. IN-GAME PURCHASE CLASSIFIER BASED ON THE SELF-ATTENTION TECHNIQUE

The original Transformer model [25] consists of two main components: an encoder and a decoder. The proposed classifier focuses only on the encoder, known for its intricate self-attention mechanism, which processes the entire input token sequence in parallel. That allows it to capture rich contextual relationships between tokens, leading to a more comprehensive data representation that uncovers short-term and long-term patterns in a player's history.

Since the tokens themselves do not have an arrangement, it is necessary to map them into some Euclidean space of meaning in which their similarity and distance could be measured, while at the same time maintaining their mutual characteristics. Embedding space is a space of vector representations of tokens in an Euclidean space that captures their underlying relationships and patterns. The process of finding the appropriate embedding space that best describes the tokens and their interconnections is done by training the TNN classification model.

The self-attention mechanism operates by computing the scalar product between each token representation vector (e_k) and other token representation vectors in a sequence of tokens ($t_1, \ldots, t_k, \ldots, t_n$) in the embedding space (see Fig. 3). The scalar product measures the similarity between two vectors within the embedding space, allowing the model to assess the significance of their relationships by comparing all pairs simultaneously. This process constructs new representation vectors (e^*_k) that represent the understanding of the entire sequence of tokens from the perspective of each token (t_k) in the input token sequences.

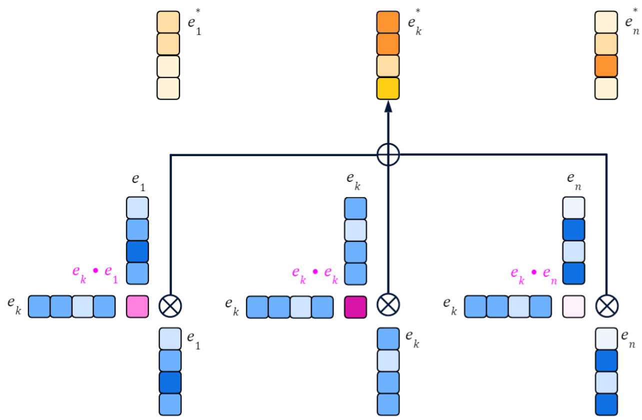

FIGURE 3. Self-Attention mechanism. The perception of the input sequence from the perspective of token t_k, where e_k is the corresponding representation vector in the embedding space. The intensity of the blue color corresponds to the numerical values within the vector representation of the tokens, and the intensity of the pink color indicates the values of the dot product, i.e., the similarity of the observed token representation vectors. Note that token t_1 is more similar to t_k than token t_n.

The encoder output contains the contextual embeddings of all tokens from the input sequence. Hence, it is applied mean pooling to obtain a single embedding vector representing the player's history. This vector is passed through an additional linear layer which outputs logits for both Purchase and No Purchase classes. The final class is selected based on the maximal logit score. The architecture of the proposed classification model is illustrated in Figure 4.

The architecture of the model is defined by selecting the embedding space size, number of multi-head attention layers in the encoder, attention heads, feedforward network hidden layer size, and other important hyperparameters such as the number of training epochs and a learning rate. The selection of these hyperparameters before the actual training will be explained in Section III-B.

III. DESCRIPTION OF THE EXPERIMENT

Four classification methods (RF, XGB, MLP, and TNN) and three history representations (Feature Statistics, Feature Tokens, and Token Frequency) were evaluated for the task of in-game purchase prediction that has a form of a classification

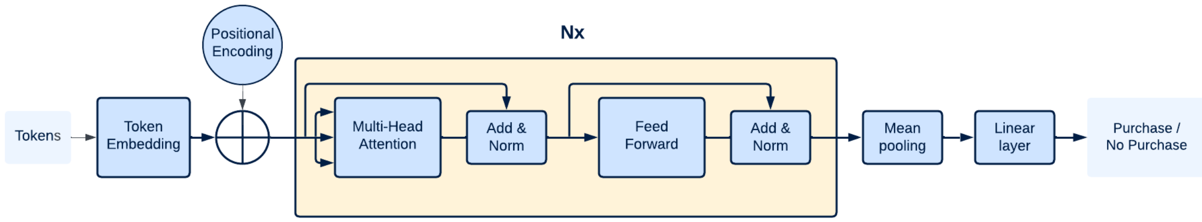

FIGURE 4. The architecture of the proposed classification model consists of *N* multi-head attention layers and Positional Encoding, which adds information about the order of input tokens. In a Self-Attention mechanism, the order of token representations is not substantial, while in a TNN-based model, it plays a significant role - tokens represent a player's behavioral history, which has a time-series nature.

problem:

$$H^p \mapsto y^p \in \{0, 1\}.$$

Seven classification models were evaluated: RF + Feature Statistics, RF + Token Frequency, XGB + Feature Statistics, XGB + Token Frequency, MLP + Feature Statistics, MLP + Token Frequency, and TNN + Feature Tokens. Each model is tested on the test part of 10 randomly generated train(80%) - test(20%) splits. The models were compared by using the standard multiclass classifiaction performance metric – macro-averaged F1 score. Since the F1 score is highly sensitive to the proportion between classes, the Area Under the ROC curve (*AUROC*) and the Matthews Correlation Coefficient (*MCC*) were also used. The *AUROC* measures the complete model's ability to distinguish between classes by evaluating the trade-off between the true positive rate and the false positive rate, making it useful for understanding model performance across different decision thresholds. The *MCC* is particularly valuable in the context of imbalanced datasets (i.e. predicting purchases in F2P games), as it provides a balanced measure of model performance by considering all four categories of predictions (true positives, true negatives, false positives, and false negatives), thus giving a comprehensive view of how well the model performs across both classes. These metrics are averaged over 10 iterations to form a final estimate.

In each iteration, the train and test parts were expanded to enhance model training on a broader dataset, and to take varying lengths of players' histories, as described in the next section.

A. EXPANDING THE INITIAL DATASET

New observations were generated for each player, covering the period from their game registration ($t = 1$) to several randomly selected points within their history ($t = T_i^p$, see Fig. 5). This approach produced nearly 240 000 observations in the training set and 62 000 observations in the test set, in each of 10 iterations (train-test splits).

B. SELECTING HYPERPARAMETERS

To select the appropriate hyperparametes, 15% of training data was extracted for validation purposes. After the best hyperparameter selections, based on the best achieved macro-averaged F1 scores on validation sets, the models are trained on a whole training set. For all models, hyperparameters were found in a grid search procedure using relevant model parameters: RF considered the number of estimators (50, 100, 200), maximum depth (10, 20, 30), and minimum samples per leaf (1, 2, 4); XGB considered the number of estimators (50, 100, 200), maximum depth (3, 6, 9), and learning rates (0.01, 0.1, 0.2). In the case of the MLP models for the input vectors of k features, hyperparameters were the number of nodes in the hidden layer ($\frac{k+1}{2}$, $2k$, $3k$), and learning rates of (10^{-3}, 10^{-2}, 10^{-1}). The output layer used a sigmoid activation function. To enhance model performance and prevent overfitting, early stopping was employed, terminating training if the validation error increased for 5 consecutive epochs.

The considered TNN model's architectures are presented in Table 4. All architectures assumed the feedforward dimension of 128. The TNN models are trained using the

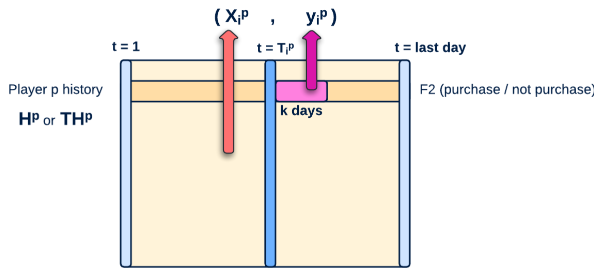

FIGURE 5. Generating new examples from a player's history matrix (H^p or TH^p).

TABLE 4. The considered model's architectures for the TNN model. Note that the token embedding size is much smaller than in the language processing tasks since the token vocabulary in our case is significantly smaller (only 48 tokens describing the player's behavioral features).

	Embedding size	Number of Heads	Number of Layers
No. 1	8	2	2
No. 2	8	4	2
No. 3	16	2	2
No. 4	16	4	1
No. 5	16	8	2
No. 6	32	2	2
No. 7	32	4	2
No. 8	32	8	2

TABLE 5. Confusion matrices and corresponding statistics for predicting purchases in the next 7 days.

Model	Confusion Matrix	F1-score	Precision for 0	Recall for 0	*AUROC*	*MCC*
RF + Feature Statistics	$\begin{bmatrix} 199\,591 & 81\,718 \\ 72\,027 & 621\,951 \end{bmatrix}$	0.8059	0.7348	0.7095	0.9028	0.6121
RF + Token Frequency	$\begin{bmatrix} 191\,968 & 89\,341 \\ 73\,397 & 620\,581 \end{bmatrix}$	0.7931	0.7234	0.6824	0.8977	0.5870
XGB + Feature Statistics	$\begin{bmatrix} 210\,055 & 71\,254 \\ 63\,832 & 630\,146 \end{bmatrix}$	0.8298	0.7669	0.7465	0.9231	0.6600
XGB + Token Frequency	$\begin{bmatrix} 191\,799 & 89\,510 \\ 73\,570 & 620\,408 \end{bmatrix}$	0.7927	0.7227	0.7465	0.8988	0.5861
MLP + Feature Statistics	$\begin{bmatrix} 173\,634 & 107\,675 \\ 61\,109 & 632\,869 \end{bmatrix}$	0.7776	0.7396	0.6172	0.8875	0.5659
MLP + Token Frequency	$\begin{bmatrix} 187\,988 & 93\,321 \\ 78\,192 & 615\,786 \end{bmatrix}$	0.7822	0.7062	0.6682	0.8604	0.5607
TNN + Feature Token	$\begin{bmatrix} 226\,110 & 55\,199 \\ 66\,925 & 627\,053 \end{bmatrix}$	**0.8495**	**0.7716**	**0.8037**	**0.9348**	**0.6989**

TABLE 6. Confusion matrices and corresponding statistics for predicting purchases in the next 5 days.

Model	Confusion Matrix	F1-score	Precision for 0	Recall for 0	*AUROC*	*MCC*
RF + Feature Statistics	$\begin{bmatrix} 225\,024 & 83\,391 \\ 73\,866 & 593\,006 \end{bmatrix}$	0.8119	0.7528	0.7269	0.9031	0.6250
RF + Token Frequency	$\begin{bmatrix} 218\,919 & 89\,496 \\ 77\,831 & 589\,041 \end{bmatrix}$	0.7995	0.7377	0.7098	0.8963	0.5994
XGB + Feature Statistics	$\begin{bmatrix} 236\,253 & 72\,162 \\ 65\,614 & 601\,258 \end{bmatrix}$	0.8357	0.7826	0.7660	0.9128	0.6715
XGB + Token Frequency	$\begin{bmatrix} 219\,220 & 89\,195 \\ 78\,426 & 588\,446 \end{bmatrix}$	0.8357	0.7826	0.7660	0.8975	0.5990
MLP + Feature Statistics	$\begin{bmatrix} 220\,402 & 88\,013 \\ 72\,559 & 594\,313 \end{bmatrix}$	0.7940	0.7523	0.7146	0.8881	0.6827
MLP + Token Frequency	$\begin{bmatrix} 216\,796 & 91\,619 \\ 80\,397 & 586\,475 \end{bmatrix}$	0.7940	0.7294	0.7029	0.8830	0.6140
TNN + Feature Token	$\begin{bmatrix} 248\,400 & 60\,015 \\ 64\,405 & 602\,467 \end{bmatrix}$	**0.8530**	**0.7941**	**0.8054**	**0.9278**	**0.7061**

Adam optimizer and the standard Binary Cross Entropy Loss function.

C. RESULTS

All performance metrics mentioned in Section III were calculated after the models were assessed using the aggregated confusion matrices with rows representing the actual 0/1 values and columns the predicted 0/1 values.The model's aggregated confusion matrix is the sum of the confusion matrices obtained from 10 iterations (train-test splits). The aggregated confusion matrices, macro-averaged F1, *AUROC*, and MCC scores for each considered period of the next k days ($k = 3, 5, 7$) are presented in the tables 5, 6, and 7.

The obtained F1 scores underline the exceptional performance of the TNN model and the Self-Attention Technique compared to traditional RF, XGB, and MLP models for all prediction periods. As expected, the models with Feature Statistics are slightly better than the models with Token Frequency since the latter approximates the distributions of feature values (the information is partially lost due to discretization). However, despite this limitation, the TNN model consistently delivers better results, underscoring the critical importance of capturing long-term patterns in the data. This highlights that a more comprehensive view of a player's history can provide significant predictive power. The confusion matrix values show that the precision for the No

TABLE 7. Confusion matrices and corresponding statistics for predicting purchases in the next 3 days.

Model	Confusion Matrix	F1-score	Precision for 0	Recall for 0	*AUROC*	*MCC*
RF + Feature Statistics	$\begin{bmatrix} 225\,367 & 83\,048 \\ 73\,788 & 593\,084 \end{bmatrix}$	0.8125	0.7533	0.7307	0.9033	0.6252
RF + Token Frequency	$\begin{bmatrix} 218\,432 & 89\,983 \\ 77\,475 & 589\,397 \end{bmatrix}$	0.7992	0.7533	0.7307	0.8962	0.5988
XGB + Feature Statistics	$\begin{bmatrix} 236\,253 & 71\,162 \\ 65\,614 & 601\,258 \end{bmatrix}$	0.8357	0.7826	0.7660	0.9128	0.6715
XGB + Token Frequency	$\begin{bmatrix} 219\,220 & 89\,195 \\ 78\,426 & 588\,446 \end{bmatrix}$	0.7993	0.7365	0.7108	0.8975	0.5990
MLP + Feature Statistics	$\begin{bmatrix} 208\,360 & 100\,055 \\ 72\,950 & 593\,922 \end{bmatrix}$	0.7897	0.7406	0.6755	0.8885	0.5819
MLP + Token Frequency	$\begin{bmatrix} 215\,296 & 93\,119 \\ 79\,677 & 587\,195 \end{bmatrix}$	0.7926	0.7298	0.6980	0.8885	0.5852
TNN + Feature Token	$\begin{bmatrix} 251\,041 & 57\,374 \\ 68\,137 & 598\,735 \end{bmatrix}$	**0.8525**	**0.7865**	**0.8139**	**0.9223**	**0.7054**

Purchase class decreases as the prediction period lengthens, whereas the precision for the Purchase class increases. This result is expected because the probability of the No Purchase event decreases as the prediction period lengthens.

In addition to achieving better F1 scores, the TNN model consistently outperforms RF, XGB, and MLP models in terms of AUROC and MCC metrics. These improvements are particularly significant given the class imbalance in the data, which is best measured by MCC. The TNN model effectively handles the skewed distribution, providing more reliable predictions across all evaluation metrics. This underscores the robustness of the proposed approach and its clear advantage over traditional models in classification. As can be seen from the tables XGB + Feature Statistics is clearly better than other classical approache showing the advantage of gradient boosting learning over other methods.

Besides the significant performance improvement, the proposed model is sensitive to the tokenization (discretization) process. Further research would concentrate on the best ways to tokenize the continual features from a player's history.

IV. CONCLUSION

Predicting purchases in free-to-play online games is one of the most critical tasks in the gaming industry. This research explores the potential of Transformer Neural Networks for in-game purchase prediction, introducing a novel approach that can significantly enhance prediction accuracy. To adapt the Transformer model for the prediction task, the study introduces a method for representing a history of play using a discrete sequence of behavioral tokens. This method leverages comprehensive data from the moment of registration and accommodates variable-length histories.

The proposed discretized representation and the associated Transformer classification model were compared against six model-representation combinations using commonly adopted methods such as Random Forest, XGBoost, and Multilayer Perceptron in predicting whether a player will purchase within the next 3, 5, or 7 days.

Although discretization methods may result in some information loss, this approach, when applied to Transformer Neural Networks, actually improves prediction accuracy. The improvement stems from the model's ability to capture long-term behavioral patterns in a player's history. Despite the loss of finer details, Transformer models can outperform commonly used machine learning techniques. Further refinement of the discretization process preceding the Transformer classifier could potentially improve model performance and offer opportunities for further enhancements.

AUTHOR CONTRIBUTIONS

Miloš A. Kovačević: Conceptulalization, Investigation, Methodology, Code development, Writing-Review, and Editing; Marko D. Pešović: Conceptulalization, Investigation, Methodology, Code Development, and Writing-Original Draft; Zoran Z. Petrović: Conceptulalization, Methodology, and Validation; and Zoran S. Pucanović: Conceptulalization, Methodology, and Project Admistration.

ACKNOWLEDGMENT

The authors would like to thank Vojislav Milutinović from the company Two Desperados, whose idea was to form their small research team. Finally, they express their gratitude to the anonymous referees for their careful reading of our manuscript and their valuable suggestions, which significantly improved this article.

REFERENCES

[1] A. Perišić and M. Pahor, "RFM-LIR feature framework for churn prediction in the mobile games market," *IEEE Trans. Games*, vol. 14, no. 2, pp. 126–137, Jun. 2022.

[2] A. Saas, A. Guitart, and Á. Periáñez, "Discovering playing patterns: Time series clustering of free-to-play game data," in *Proc. IEEE Conf. Comput. Intell. Games (CIG)*, Sep. 2016, pp. 1–8.
[3] C. Bauckhage, A. Drachen, and R. Sifa, "Clustering game behavior data," *IEEE Trans. Comput. Intell. AI Games*, vol. 7, no. 3, pp. 266–278, Sep. 2015.
[4] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in tomb raider: Underworld," in *Proc. IEEE Symp. Comput. Intell. Games*, Sep. 2009, pp. 1–8.
[5] N. Chaudhuri, G. Gupta, V. Vamsi, and I. Bose, "On the platform but will they buy? Predicting customers' purchase behavior using deep learning," *Decis. Support Syst.*, vol. 149, Oct. 2021, Art. no. 113622.
[6] G. Chaubey, P. R. Gavhane, D. Bisen, and S. K. Arjaria, "Customer purchasing behavior prediction using machine learning classification techniques," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 12, pp. 16133–16157, Dec. 2023.
[7] W. Yang, T. Huang, J. Zeng, Y. Tang, L. Chen, S. Mishra, and Y. E. Liu, "Purchase prediction in free online games via survival analysis," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 4444–4449.
[8] A. Grigoras and F. Leon, "Transformer-based model for predicting customers' next purchase day in e-commerce," *Computation*, vol. 11, no. 11, p. 210, Oct. 2023.
[9] Y. B. Kim, K. Kang, J. Choo, S. J. Kang, T. Kim, J. Im, J.-H. Kim, and C. H. Kim, "Predicting the currency market in online gaming via lexicon-based analysis on its online forum," *Complexity*, vol. 2017, pp. 1–10, Jun. 2017.
[10] J. Runge, P. Gao, F. Garcin, and B. Faltings, "Churn prediction for high-value players in casual social games," in *Proc. IEEE Conf. Comput. Intell. Games*, Aug. 2014, pp. 1–8.
[11] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, Feb. 1970.
[12] H. Xie, S. Devlin, D. Kudenko, and P. Cowling, "Predicting player disengagement and first purchase with event-frequency based data representation," in *Proc. IEEE Conf. Comput. Intell. Games*, Aug. 2015, pp. 230–237.
[13] R. Sifa, F. Hadiji, J. Runge, A. Drachen, K. Kersting, and C. Bauckhage, "Predicting purchase decisions in mobile free-to-play games," in *Proc. AAAI AIIDE*, 2015, pp. 79–85.
[14] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[15] E. Endriss and J. Leite, "Predicting players behavior in games with microtransactions',' in *Proc. 7th Eur. Starting AI Researcher Symp.*, vol. 264, 2014, p. 230.
[16] R. Sifa, F. Hadiji, J. Runge, A. Drachen, K. Kersting, and C. Bauckhage, "Predicting purchase decisions in mobile free-to-play games," in *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2021, vol. 11, no. 1, pp. 79–85.
[17] F. Hadiji, R. Sifa, A. Drachen, C. Thurau, K. Kersting, and C. Bauckhage, "Predicting player churn in the wild," in *Proc. IEEE Conf. Comput. Intell. Games*, Aug. 2014, pp. 1–8.
[18] P. Bertens, A. Guitart, and A. Perlanez, "A scalable multi-dimensional churn prediction," in *Proc. IEEE Conf. Comput. Intell. Games (CIG)*, 2017, pp. 33–36.
[19] V. Lazarov and M. Capota, "Churn prediction," *Bus. Anal. Course. Comput. Sci.*, vol. 33, p. 34, Jun. 2007.
[20] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, and S. W. Kim, "A churn prediction model using random forest: Analysis of machine learning techniques for churn prediction and factor identification in telecom sector," *IEEE Access*, vol. 7, pp. 60134–60149, 2019.
[21] Y. Xie, X. Li, E. W. T. Ngai, and W. Ying, "Customer churn prediction using improved balanced random forests," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5445–5449, Apr. 2009.
[22] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.
[23] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinf.*, vol. 8, no. 1, pp. 1–21, Dec. 2007.
[24] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinf.*, vol. 9, no. 1, Dec. 2008.
[25] V. Ashish, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, p. 30.
[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, Dec. 2014.
[27] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," 2017, *arXiv:1703.03130*.
[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[29] J. Rabbah, M. Ridouani, and L. Hassouni, "New approach to telecom churn prediction based on transformers," in *Proc. Int. Conf. Artif. Intell. Comput. Vis.*, 2023, pp. 565–574.
[30] G. Box, "Jenkins: Time series analysis, forecasting, and control," in *A Very British Affair: Six Britons and the Development of Time Series Analysis During the 20th Century*. 2013, pp. 161–215.
[31] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, vol. 11, Aug. 2016, pp. 785–794.
[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[33] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[34] N. A. Weiss and C. A. Weiss, *Introductory Statistics*. London, U.K.: Pearson, 2017.

MILOŠ A. KOVAČEVIĆ received the Ph.D. degree in computer science from the School of Electrical Engineering, University of Belgrade, Serbia, in 2007. He is currently working as a Full Professor of computing sciences and computer engineering with the Faculty of Civil Engineering, University of Belgrade. His research interests include text mining and applications of machine learning in various engineering disciplines.

MARKO D. PEŠOVIĆ received the Ph.D. degree in algebraic combinatorics from the Faculty of Mathematics, University of Belgrade, Serbia, in 2021. He is currently working as an Assistant Professor of mathematics with the Faculty of Civil Engineering, University of Belgrade. His research interests include algebraic combinatorics, linear algebra, statistics, and machine learning.

ZORAN Z. PETROVIĆ received the Ph.D. degree in algebraic topology from the Johns Hopkins University, Baltimore, USA, in 1996. He is currently working as a Full Professor of algebra and mathematical logic with the Faculty of Mathematics, University of Belgrade, Serbia. His research interests include algebraic topology, commutative algebra, ring theory, linear algebra, and machine learning.

ZORAN S. PUCANOVIĆ received the Ph.D. degree in algebra from the Faculty of Mathematics, University of Belgrade, Serbia, in 2013. He is currently working as an Associate Professor of mathematics with the Faculty of Civil Engineering, University of Belgrade. His research interests include ring theory, numerical linear algebra with applications, discrete mathematics, and machine learning.