Multi Datasource LTV User Representation (MDLUR)*

Junwoo Yun*
Bagelcode
Seoul, Korea
junwoo.yun@bagelcode.com

Wonryeol Kwak*
Bagelcode
Seoul, Korea
wonryeol.kwak@bagelcode.com

Joohyun Kim
Bagelcode
Seoul, Korea
joohyun.kim@bagelcode.com

ABSTRACT

In this paper, we propose a novel user representation methodology called Multi Datasource LTV User Representation (MDLUR). Our model aims to establish a universal user embedding for downstream tasks, specifically lifetime value (LTV) prediction on specific days after installation. MDLUR uses a combination of various data sources, including user information, portrait, and behavior data from the first n days after installation of the social casino game "Club Vegas Slots" developed by Bagelcode. This model overcomes the limitation of conventional approaches that struggle with effectively utilizing various data sources or accurately capturing interactions in sparse datasets. MDLUR adopts unique model architectures tailored to each data source. Coupled with robust dimensionality reduction techniques, this model succeeds in the effective integration of insights from various data sources. Comprehensive experiments on real-world industrial data demonstrate the superiority of the proposed methods compared to SOTA baselines including Two-Stage XGBoost, WhalesDector, MSDMT, and BST. Not only did it outperform these models, but it has also been efficiently deployed and tested in a live environment using MLOps demonstrating its maintainability. The representation may potentially be applied to a wide range of downstream tasks, including conversion, churn, and retention prediction, as well as user segmentation and item recommendation.

CCS CONCEPTS

• Computing methodologies → Machine learning; **• Information systems → Expert systems**.

KEYWORDS

user representation, LTV prediction, sparse data embedding, dimensionality reduction, purchase prediction, mobile game

ACM Reference Format:
Junwoo Yun, Wonryeol Kwak, and Joohyun Kim. 2023. Multi Datasource LTV User Representation (MDLUR). In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3580305.3599871

*Real-world application with efficient dimensionality reduction techniques

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08...$15.00
https://doi.org/10.1145/3580305.3599871

1 INTRODUCTION

In the rapidly evolving gaming industry, companies continuously strive to enhance their revenue streams and improve player acquisition. Therefore, understanding player behavior has become a strategic necessity. The growing focus on LTV, which quantifies the revenue a player generates until they permanently leave the game, underscores this point. By leveraging insights derived from data and LTV, companies can not only increase revenue but also foster improved gaming experiences through personalized purchases [1, 30]. Furthermore, they can implement marketing strategies to identify and attract more valuable players by identifying the similarities in demographics and behaviors they share with existing valuable users.

As player-generated data grows, game companies are in need of sophisticated prediction models to better understand and forecast player purchases. This leads to more targeted and effective in-app purchase (IAP) strategies. Specifically on LTV, conventional approaches such as RFM and BYTD [7, 8, 10–12, 16, 17, 25, 27], machine learning [6, 19, 22], and imbalance fixation [19, 20, 24] have been utilized. However, these primarily focus on purchases and directly related behaviors, require additional feature selections, and may not holistically capture the relationships within the data. The advent of deep learning has facilitated a deeper comprehension of user behavior by using behavior sequence data [3, 4, 32] and user representations [5, 18, 26, 29, 31].

While existing methods demonstrate potential in improving LTV prediction, they primarily focus on recommendation systems for e-commerce data and associated downstream tasks. Studies related to the gaming industry [1, 2, 5, 19] do not fully utilize the diversity of game data sources and lack a comprehensive understanding of these. Recent methods have begun to consider a broader range of data features and sources, including M3Rec [4], and SCGRec [30]. Nevertheless, these methods still fall short of fully using different sources and methods to comprehend relationships within the data. The reason for the use of limited features might be due to the trade-off of employing large feature sets from multiple sources, which often results in the curse of dimensionality due to data sparsity.

To address these challenges, we propose the Multi Datasource LTV User Representation (MDLUR). This novel approach establishes a universal user embedding for downstream tasks, particularly LTV prediction on a specific day after the installation. The novelty of MDLUR lies in its capacity to utilize multiple data modalities and spectrums, adopting distinct and unique model architectures tailored to each data source – user information, portrait, and behavior sequence data. The output representations from each source are aggregated and condensed via a Skip-connected Autoencoder (SAE), resulting in a rich and robust user representation that encapsulates various aspects. This approach enables our model to excel in downstream tasks, notably LTV prediction, offering a deeper understanding of player behavior.

Furthermore, MDLUR, through its universal user embedding, offers significant advantages over traditional models. It provides a robust estimator for various downstream tasks and uncovers potential in underrepresented players that conventional models often overlook. By adopting a holistic approach to user, portrait, and behavior data, MDLUR boosts company revenues and enhances the overall player experience.

While utilizing multi-datasource with large sparse embedding may lead to the curse of dimensionality, we adopted efficient dimensionality reduction techniques, including Conv2D, Autoencoder (AE), SAE, and UNET architectures, supplemented by appropriate scaling and initialization. As a result, our model generates compact embeddings applicable to various downstream tasks, including conversion, churn, and retention prediction, as well as user segmentation and item recommendation. Comparisons with baseline models and offline experiments have shown the superiority of MDLUR. Moreover, it has been efficiently deployed and online-tested in a live production environment via MLOps, which demonstrates its maintainability.

2 RELATED WORK

2.1 Deep Learning

Recent advancement in deep learning techniques has enhanced LTV prediction, compared to classical approaches such as statistical RFM [7, 8, 23] and Boosting models [6]. Chen et al. [2] was the first to introduce DNN to tackle the problem. It outperformed boosting models in terms of accuracy and had the ability to train on larger sequential data without feature engineering, thereby significantly reducing computation time and resources.

In response to shortcomings such as class imbalance and payer distribution, Wang et al. [24] proposed a modification to the model introduced by Chen et al. [2]. This study assumes that the payer distribution follows a Zero-Inflated Log-Normal (ZILN) distribution, shown in equation 1, and employs DNN to fit the mean and standard deviation of the distribution for LTV prediction. They introduced a loss function to handle the long-tail skewed distribution. We have adopted and tested this strategy with our MDLUR framework due to similar challenges in our data. A recent work [14] introduced an industrial-grade model capable of handling diverse distributions for LTV prediction, which emphasizes the importance of data distributions.

$$L_{ZILN}(x;p,\mu,\sigma) = -\mathbb{I}_{\{x=0\}} log(1-p) - \mathbb{I}_{\{x>0\}}(log p - L_{Lognormal}(x;\mu,\sigma)) \quad (1)$$

The class imbalance was further addressed by using Synthetic Minority Over-sampling Technique (SMOTE) in conjunction with DNN to predict premium and high-value users [18, 19]. This approach can adapt to a variety of data not directly related to the purchase and explicitly handles the high imbalance and skewness prevalent in freemium game datasets. To capture the temporal sequences and their long-range dependencies within a user's purchase history and gameplay action, del Río et al. [5] employed Long Short-Term Memory (LSTM) for LTV prediction.

TSUR [28] leveraged the social network along with time series revenue history, employing a Graph Attention Network and Wavelet Transform with Multi-headed attention. This architecture successfully learns both temporal and structural user representations in the low-pass representation space, effectively filtering data noise for LTV prediction.

2.2 Sequential Portrait and Behavior Model

Distinct from traditional approaches that strictly focus on user and purchase-related data, more recent studies have begun to consider a broader spectrum of indirect or non-purchase-related events, such as user's portrait and behavior data [34]. This study employs user behavior feature embedding to establish a relationship between user action and target prediction with local activation, supplemented by fully connected layers in the embedding. BST [3] implemented a transformer architecture on chronologically ordered user behavior sequences with LeakyReLU and dropout, capturing interactions in sparse behavior features and understanding the sequential nature of user behavior. Our proposed MDLUR further enhances this approach by adopting Time2Vec [13] for better time embedding and the UNET architecture for efficient dimensionality reduction. Aggregating spatial representation from the transformer layers and the UNET with a weighted sum resulted in better performance, taking advantage of both approaches.

The recent model PSAC [32] has demonstrated improved prediction accuracy by leveraging the interconnections between multiple data sources. The paper employs N_Gram embedding on sequential behavior data, alongside DNN and LSTM layers for purchase prediction. M3Rec [4] utilized sequential gaming data with user levels based on varying action types via a Graph Neural Network for better user comprehension. Our model expands upon this by diversifying the data sources and tailoring a specific architecture to each source, which maximizes information extraction and depth of understanding.

2.3 User Representation

Different sources of data provide implicit and explicit information. Construction of user representations [3, 34] not only aids in predicting LTV but can also be fine-tuned or transferred to other downstream tasks. Therefore, an efficient user representation is crucial for both an in-depth understanding of users and downstream prediction tasks.

Recent studies [18, 31] propose a general-purpose representation learning approach through large-scale pre-training within the e-commerce domain. This approach can be utilized for various downstream tasks, including LTV prediction. A study by Yang et al. [29] further enhanced universal user representation through the spatial encoding of user behavior and Self-supervised Multi-anchor Encoder Network (SMEN), which develops multiple low-dimensional user representations through contrastive learning. The work by Wu et al. [26] treats sequential user representation similar to an NLP task, employing the BERT architecture to establish the relatedness between user behavior and sequence matching, resulting in improved performance. Within the gaming domain, only a few studies [14, 28] have employed representation learning with limited use of the data and less efficient model architectures.

Our model sets itself apart from previous studies in the gaming industry by leveraging data streams from multiple perspectives, including user information, in-game portrait, and behavior data. The employment of distinct model architectures for each data stream enhances generalization performance, as each architecture captures unique aspects of the data. Weighted-sum aggregation allows the model to assign the importance of information from each data source. Consequently, our MDLUR delivers rich and robust universal user representations for LTV prediction tasks, a novel stride for enhanced prediction results and further downstream tasks.

3 DATA AND ANALYSIS

The effectiveness of MDLUR comes from its use of multiple data sources. Recent studies [3, 4, 18, 29] have demonstrated the value of diverse data sources, which improves the understanding of feature interactions for predictive tasks. We focus on the use of larger spectra of the data streams from different perspectives to fully understand the users with three distinct data sources - user information, portrait, and behavior data. The unique architecture handles each data source separately and later concatenates them to create a robust and comprehensive representation. The rationale for the separation lies in the unique perspectives offered by each data type, providing a multifaceted understanding of the user that a single data source might miss.

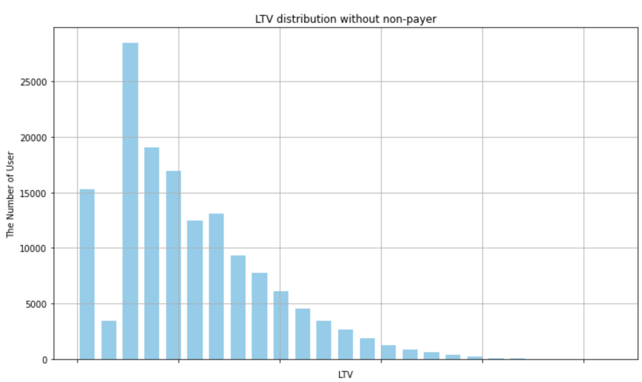

Figure 1: Distribution of the LTV per users

Table 1 underscores that only 3% of the total user base is responsible for generating the entire LTV. Given this significant imbalance, the crucial part lies in the model's capacity to accurately distinguish between paying and non-paying users and to predict the purchase volume of paying users precisely. Furthermore, Figure 1 reveals a notable sparsity among high LTV users, creating a skewed asymmetric long-tail distribution problem that complicates predicting user purchases and amounts. From efficient data sampling suggested by [14, 19, 20, 24], we under-sampled at a 1:3 ratio of payers to non-payers, and filtered out inactive users who did not play game sufficiently after installation. This approach has been fine-tuned to yield the best results.

Table 1: Statistics of the data

Statistics	Values
Date Range	2022-01-01 to 2022-12-31
Number of the User	4,470,444
Payer - Non Payer ratio	1 : 30
User Information	12 fields
Portrait	248 static, time-series fields
Behavior	165 time-series fields

3.1 General Information and Statistics

The data used in this paper was collected from the mobile game "Club Vegas Slots", developed by Bagelcode, a global mobile publisher with over 50 million users worldwide. It comprises 4.5M anonymized user datasets who joined the game between 2022-01-01 and 2022-12-31. All data in this paper were used with the users' consent to the Privacy Policy.

Table 2: Feature category of each source of the data

Category / Source	User Info	Portrait	Behavior
Base	user info	purchase, login, level/tier up, monetization	spin, purchase, login, reward, mission, monetization
Social		epic elbum, share, friend, club	chat, share, friend, club
Popup		popup, IAM	popup, IAM
Click		click	click
Enter		enter, slot enter	enter, slot enter

We conducted grid-search to select the most appropriate features for our model, after grouping features from each source into five categories: base, social, popup, click, and enter, as shown in Table 2. Notably, features in the "popup" category partially overlap with the "base" category, leading us to remove them to lessen sparsity and enhance model efficiency. In total, we used 12 fields of user information, 248 fields of portrait (static and time-series), and 165 fields of behavior purchase data. Given that the total number of features exceeds 400, including rare in-app events, data sparsity is anticipated. This sparsity, which varies by user, is a primary consideration in the novel approach of MDLUR to address this challenge.

3.2 User Information

The user information data encompasses static data primarily related to the users, particularly focusing on attribution details such as source of installation. As the company seeks to attract high-value users through targeted marketing efforts, understanding these attributes with demographic plays a main role in understanding the users.

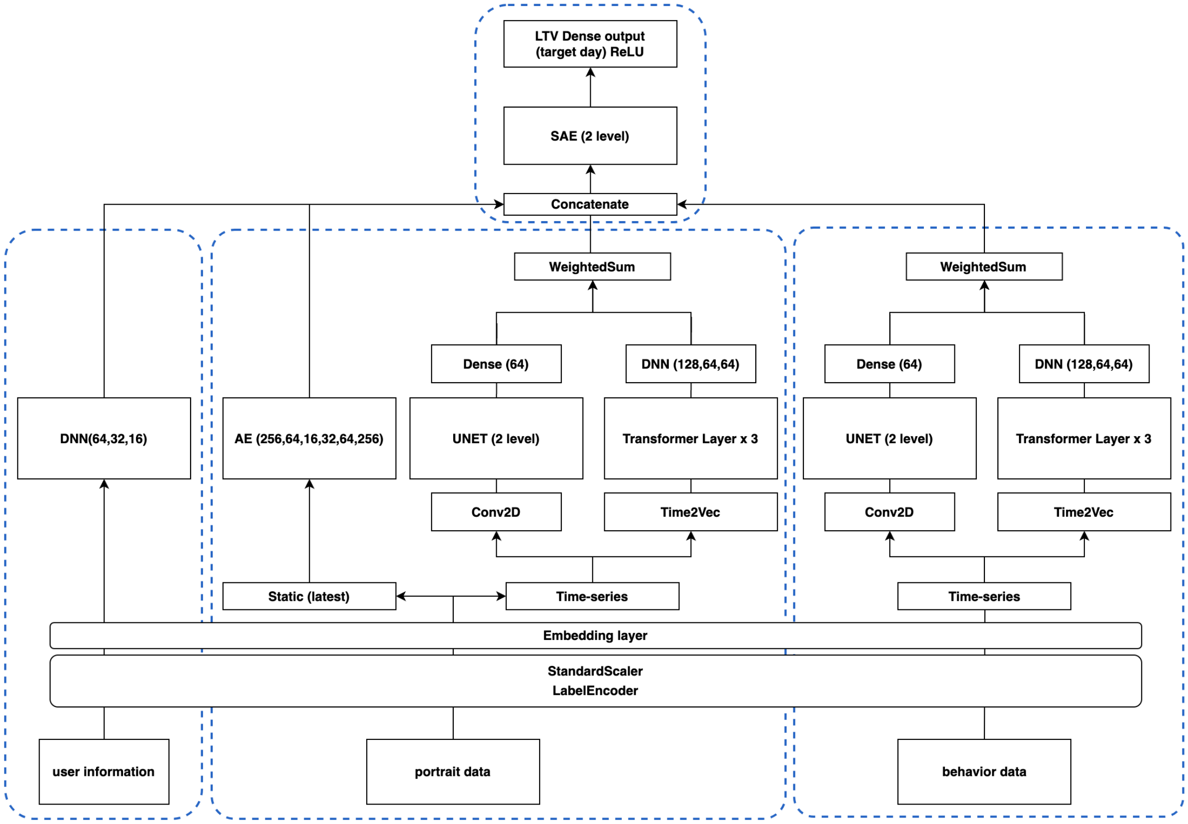

Figure 2: The overview architecture of the proposed model. MDLUR takes user information, portrait data, and behavior data in sequence. All data sources are first preprocessed with data cleanser, scaler and encoder, then embedded into a low-dimensional vector. Distinct approaches are employed for each data source incorporating dimensional reduction techniques, which is visually distinguished by dotted lines. The outputs from each source are then concatenated with a two-level SAE to learn the interaction of hidden features while managing sparsity. The final output is generated using the DNN with ReLU activation.

3.3 Portrait Data

Portrait data represents user status in the form of static and time-series data. The time-series data represents the evolution of the user's status since installation, while the static data only preserves the most recent status, given its importance in predicting future LTV. As outlined in prior research [3], three aggregation functions were used on the static data to track changes in the user's portrait status: cumulative sum, difference, and percentage difference. The CNN and Transformer layers within the time-series data architecture would further track the changes in the data. The combination of static and time-series data furnishes a rich dataset, enabling the model to understand the significance of a user's status and its implications for downstream tasks.

3.4 Behavior Data

Behavior data represents user actions within the game, presented in a time-series format. This data is segmented into 28 windows, each representing aggregated data over a specified time frame. For instance, given 7 days of input data, each window encapsulates 6 hours of user activity. The number of windows was fine-tuned and could be adjusted to ensure optimal results and to suit specific needs. The use of aggregation with Time2vec [13] embedding allows the model to reveal patterns and trends that may not be immediately apparent in the raw data.

3.5 Summarization

From 4.5M users registered within 2022, data were collected across different sources: user information, portrait, and behavior. A total of 12 fields of user information data, 248 fields of portrait static data, 70 fields of portrait time-series data, and 165 fields of behavior purchase data were used.

The statistics highlighted the challenge of the work: the sparsity of high LTV users, the long-tail distribution of non-payers (0 LTV), and the sparse data resulting from a huge number of features. The model architecture and sampling techniques are designed to address these challenges inherent in the data.

These data categories provide a comprehensive understanding of the various situations that can occur in the game. User data supports the reasoning of downstream tasks with demographics and UA information, behavior data captures the actions taken by the user, and portrait data captures the conditions and the context of these actions.

4 METHODS

4.1 Introduction

The proposed MDLUR model architecture [1] exhibits novelty in its approach towards different data sources, owing to the distinct information and perspectives each source offers about the users. Given the large number of features, the model incorporates dimensionality reduction techniques within the architecture. This efficiently removes the noise and captures relevant information. The unique architectures for each source are selected based on the ablation study and concatenated later to jointly learn the user representations for downstream tasks efficiently.

4.2 Preprocess

In our research, we processed data cleansing, encompassing sanity checks and outlier filtration to maintain data integrity and enhance its overall quality. These measures ensure our model is supplied with robust and uncontaminated data sources, contributing to the reliability of our findings.

Given the inherent sparsity of embeddings across multiple low-dimensional features, we recognize the importance of scaling values within the features. By ensuring uniform scales across all features, our model is able to efficiently learn the interactions between features and could be used for different downstream tasks.

4.3 Architecture

One of the key components of the MDLUR architecture is the use of weight initialization techniques to alleviate the sparsity problem and enhance model performance. The architecture consists of the LeakyReLU activation function with the He-normal kernel, which is designed to address the gradient vanishing problem caused by sparse data. The He-normal initialization normalizes weights to maintain the variance of activations close to one which prevents the gradients from becoming excessively small during training due to sparsity. As [9] contends, He-normal is an optimal match with LeakyReLU. This is proven by our experiments, demonstrating a better prevention of gradient vanishing and exploding gradient problems than other initialization methods including Xavier, Random, or uniform initialization.

In the proposed architecture, we employed distinct architectures to integrate each data source, as illustrated in Figure 2. The tailored model architecture for each source is the key distinction from previous approaches. These models consist of DNN, AE, UNET, and Transformer layers, which are later concatenated using a weighted sum and a SAE. The overall architecture is built to maximize information gain and encapsulate various aspects.

4.3.1 User Information Model Architecture. The first model architecture uses user information with a simple DNN (64, 32, 16), similar to the approach utilized in WhalesDetector[2]. This data is simple and has a small number of features, thus DNN with Dropout (0.25) is sufficient to extract important values. The output representation

[1] MDLUR is publicly available on GitHub repository

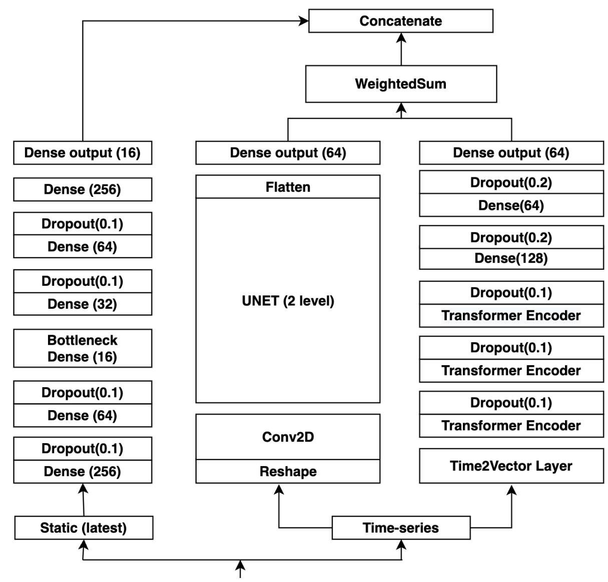

Figure 3: Overall architecture of portrait data in static and time-series format. Static data fed into AE. Time-series data fed into Conv2D – UNET, and Time2Vec - three layers of Transformer encoder – DNN. Two outputs from time-series data are concatenated using a trainable weighted sum layer.

in terms of user information provides meaningful and distinct context when it is later aggregated with other sources which enhances the overall performance.

4.3.2 Portrait Model Architecture. The second model utilizes portrait data, divided into static and time-series data. Given the dynamic nature of user status, it is crucial to consider both the time-series (status changes over time) and static (the most recent status) data. Therefore, the data is divided and modeled independently as shown in Figure 3.

The static portrait data passes through an AE (256, 64, 16, 32, 64, 256) structure for dimensionality reduction. Given that it contains 248 features, AE would filter out noise and irrelevant information from the input data by training the model to reconstruct only the most important features. This ensures stability of the model from shifting data trends, and free from the need for additional feature engineering.

The time-series portrait data uses two distinct model structures. The first is comprised of Conv2D, UNET with 2 levels, and Dense (64) output layer. The UNET encoder is composed of two Conv2D and BatchNorm layers with MaxPool2D, while the decoder is composed of Conv2DTranspose, Zero padding followed by two Conv2D and BatchNorm layers as illustrated in Figure 4.

The second structure is composed of a Time2Vec layer [13] for better time embedding, three transformer encoder layers with multi-headed attention units followed by DNN (128, 64), and a Dense (64) output layer. The outputs from each structure are concatenated with a trainable weighted sum layer.

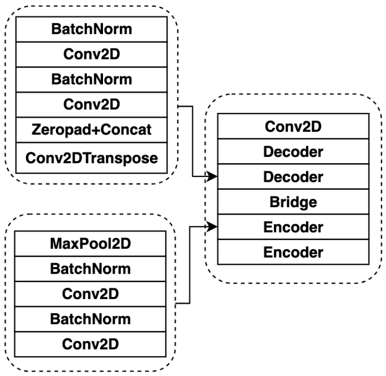

Figure 4: Architecture of two-level UNET used in time-series portrait data. The architecture is same for the behavior data.

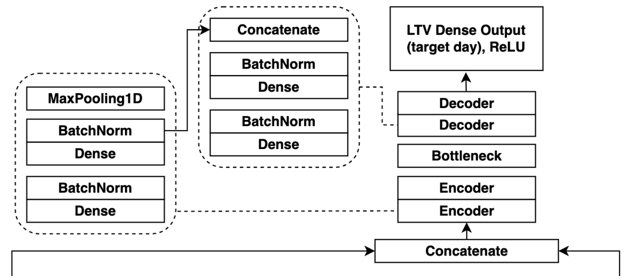

Figure 5: Architecture for concatenation, two-level SAE with output Dense layer with size specific to the task

The first structure emphasizes dimensionality reduction, tailored for handling huge and sparse portrait data. The UNET architecture leverages skip connections between the encoder and the decoder, preserving spatial information while eliminating noise from the data. The second structure focuses more on capturing status changes for better train patterns and regularities of sequential data with self-attention mechanisms. This enables the model to capture temporal dependencies, understanding how the status influences the user purchase decision-making process. The advantages from different architectures are maximized with the weighted sum, allowing the model to optimize the combination of these output vectors and show better performance than the single architecture.

4.3.3 Behavior Model Architecture. The third model architecture utilizes behavior data and has the same architecture as the portrait time-series data. The UNET architecture, applied to behavior data, derives additional benefits from its encoder-decoder structure. Regenerating chronological user behavior aids in comprehending decision-making patterns, where UNET implicitly analyzes behavior patterns and preferences leading to the purchase. Additionally, transformer layer extracts understanding of complex interaction between sequential behavior actions through multi-headed self attention. Both architectures are tailored to efficiently identify and learn behavior patterns, which enhances the predictive power on downstream tasks.

4.3.4 Concatenation Architecture. For the final user representation, outputs from all three architectures are merged and processed through a two-level SAE contrasts to conventional methods of employing a separate concatenation function. As SAE additionally uses a skip connection from AE, it harnesses both low-level and high-level features with inherent characteristics, prevents overfitting, and conducts dimensionality reduction by filtering out noise without information loss. This provides seamless integration of insights derived from diverse data sources, enhancing the robustness and accuracy of the final user representation. The architecture is shown in Figure 5 with detailed component layers of the SAE.

The output dimension depends on the prediction task, different from traditional studies that focus solely on predicting LTV at a specific date. For instance, when predicting LTV on Day 14, a ReLU Dense layer with size 14 is used for the output containing LTV from Day 1 to 14. We utilized the available y_true provided in the input data to leverage the advantages of partial target leakage. It provides insights into the temporal dependencies and patterns of changes in LTV throughout the day. This approach resulted in additional aids on the model performance and stability.

5 EXPERIMENTAL EVALUATION

5.1 Experimental Settings

5.1.1 Evaluation Dataset. We have conducted a performance evaluation of the model, alongside qualitative insights. All experiments were conducted on an Amazon EC2 p3.2xlarge instance using Tensorflow. To ensure comprehensive analysis, we separated the experiments into baseline model comparisons, additional input and target tests, and offline tests. We employed data from the x days right after installation to predict the LTV at y days, where y is always larger than x. This approach can assess effectiveness of the model in identifying more valuable users in an early period, which is a crucial component in marketing strategy.

With 12 months of data collected, we created 10 fold time-series cross-validation sets for the credibility of the results. The data in each fold was split chronologically into train, validation, and test sets to prevent target leakage in the time-series environment. We conducted a grid-search test to identify the most cost-efficient splitting without compromising the accuracy: 60 days of train, 10 days of validation, and 7 days of test data were selected.

Additionally, the data was sampled based on purchase behavior. According to the studies [20, 24], applying imbalance fixation techniques is necessary to overcome the imbalance between payer and non-payer predicting LTV, such as SMOTE, under-sampling, over-sampling, tailored loss function, and specific metrics. We adopted under-sampling to training and validation data with a 1:3 payer to non-payer ratio as determined through grid search experiments. The test data was kept unsampled to maintain its representativeness of all user groups in real-world conditions.

5.1.2 Evaluation Metrics. We adopted both regression and classification evaluation metrics for a comprehensive assessment of model efficiency. Following [22], we employed Root Mean Square Error (RMSE) and R-squared (R2) as regression metrics, given their wide acceptance for evaluating regression model. Additionally, classification metrics including Weighted Average Precision (WAP), Recall

Table 3: Result of the study, predicting Day 14 LTV from training 7 days of data from the installation

Methods	RMSE	R2	WAP	WAR	WAF1
RFM - Pareto/NBD [7]	77.41	0.29	0.33	0.36	0.32
RFM - BG/NBD [8]	64.65	0.33	0.27	0.41	0.33
RFM - MBG/NBD [23]	61.26	0.31	0.35	0.33	0.34
Two-stage XGBoost [6]	65.54	0.42	0.58	0.49	0.51
WhalesDetector [2]	43.32	0.51	0.61	0.38	0.49
ZILN [24]	170.23	-0.21	0.41	0.12	0.19
MSMDT [33]	32.24	0.59	0.65	0.59	0.63
BST [3]	30.19	0.65	0.67	0.63	0.64
MDLUR – ZILN	48.23	0.23	0.43	0.27	0.32
MDLUR – MSLE	**28.13**	**0.84**	**0.85**	**0.82**	**0.83**

(WAR), and F1-score (WAF) were employed. We transformed the LTV into five groups, with a bin of [-1, 0.5, 10, 100, 1000, inf], to calculate the classification metrics. These bin thresholds were selected through internal standards of user segmentation. This binning approach allows us to cast the regression problem into a classification problem, providing an additional lens on evaluation.

5.1.3 Evaluation Hyper-Parameter Tuning. We conducted a grid-search test to identify the most suitable loss function and hyper-parameters, considering target distributions and sparsity of the data.

Table 4: Result of testing different loss on our model MDLUR predicting LTV D14

Loss	MAE	R2	RMSE
MSE	42.24	0.68	32.89
MAE	88.67	0.62	46.46
ZILN	280.24	-0.21	160.24
MAPE	40.56	0.67	41.28
MSLE	18.32	0.74	31.21

For loss functions, we evaluated Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Squared Logarithmic Error (MSLE), Mean Absolute Percentage Error (MAPE), and Zero-Inflated Log-Normal (ZILN) functions, referencing Wang et al. [24]. As shown in Table 4, we identified MSLE as the most effective loss function due to the logarithmic characteristics to handle rare payer and skewed LTV distributions, shown in Figure 1.

For the hyperparameters, we chose ADAM optimizer with a learning rate of 0.001. For the learning rate, we tested 0.01, 0.05, 0.001, and 0.0005, and 0.001 showed continuous validation loss decay and the best metrics. Additionally, we adopted the batch size of 512 and epochs of 50 with callbacks of ReduceLROnPlateau and EarlyStopping to generalize the model and efficiently obtain the global minima.

5.2 Baseline and Evaluation Metrics

5.2.1 Baseline model comparisons. We compared our model against the following baseline models, using data from the 7 days right after installation to predict the LTV at 14 days. To reproduce these baseline models, we used publicly available libraries and code from GitHub, cited in the original paper or published by the author.

- RFM [7, 8, 23] uses Recency, Frequency, and Monetary value of the purchase with three parametric models with different distribution assumptions on RFM: Pareto/NBD, BG/NBD, MBG/MBD. Note that RFM only utilizes recency, frequency, monetary value of the purchase and disregards the rest of the data such as user information, portrait, and behavior.

- Two-stage XGBoost [6] uses a two-step process for LTV prediction. It first estimates whether a user is a payer or a non-payer, and subsequently predicts the LTV of the user.

- WhalesDetector [2] uses a three-layer CNN (300, 150, 60 nodes with conv-pool) followed by a kernel size (7, 3, 1) to detect whether the user is a high payer (whale). We reproduced it as a regression model using ReLU at the output layer to predict LTV.

- ZILN [24] uses ZILN (zero-inflated long-tailed) loss on the WhaleDetector model architecture, estimating that a user purchase follows a log-normal distribution with zero-inflation.

- MSDMT [33] utilizes heterogeneous multi-datasource, including player portrait tabular data, behavior sequence data, and social network graph data, which leads to the comprehensive understanding of each player. Since our data does not include social network information, we employed the model excluding GNN which consists of player portraits with LSTM layers + behavior sequence with CONV-1D followed by LSTM layers and concatenated by FC layers.

- BST [3] uses a transformer architecture with LeakyReLU and dropout on behavior sequence data of the user to capture interactions in sparse dataset.

We performed training five times for each model to mitigate the impact of potential local optimization problems or overfitting/underfitting. Any individual result that differed by more than 20% from its mean was considered an outlier and excluded from the final results. In each training, we conducted 10-fold time-series cross-validation across different date ranges to get generalized results. All metrics were consistently reported to two decimal places for ease of comparison.

5.2.2 Result and Analysis. As shown in Table 3, MDLUR outperforms all baseline models across all five metrics. In specific, parametric models and simple DNN [2, 6–8, 24] could not effectively capture meaningful information within sparse and complex multi-datasources. ZILN loss, argued to be effective in capturing LTV by [24], showed poor performance both in DNN and our model architecture. While CNN-LSTM used in MSDMT [33] helps to address the data sparsity and capture underlying patterns and long-term dependencies, it struggles to predict the user's purchases accurately. Similarly, BST [3] captures the sequential signal underlying user behavior sequences using transformer, but it faces difficulties in concatenating various data sources due to the curse of dimensionality.

MDLUR's superiority lies in its distinct structure, which separates and tailors model architecture to different data sources to efficiently captures the information from various aspects. Additionally, each part of the model has a key-part for dimensionality reduction, which enables efficient training on huge and sparse multi-datasources. This approach acknowledges the importance of understanding diversity in data and avoids reliance on a single complex model. Instead, it utilizes distinct, efficient models for each data source and seamlessly integrates them, creating an effective, multifaceted model for diverse downstream tasks. It is supported by state-of-the-art performance on tasks such as LTV prediction.

5.3 Experiments on Additional Input and Target Result

For an in-depth performance verification, we evaluated our proposed model with different input and target lengths, as shown in Table 5. When predicting a longer period of LTV, the Day 28 LTV prediction task result showed the model's strengths, using longer input horizons. While it showed higher RMSE and relatively less R2, it is acceptable since Day 28 LTV has higher mean and variance among the experiments. Conversely, when predicting the same target with a shorter input horizon, the model showed relatively weaker performance but within the industrially acceptable range. For instance, it is shown that twice the length of input horizons would efficiently predict the future LTV, and the longer the input data, the better the prediction.

Table 5: Result of the additional study, predicting Day 7, 14, 28 LTV from Day 1, 3, 7, 14 input data from the install. Note that the Day 7-Day 14 LTV is the base experimental setting of MDLUR

Input \ Target	RMSE	R2	WAP	WAR	WAF1
Day 1 \ Day 7 LTV	28.7	0.39	0.41	0.45	0.34
Day 3 \ Day 7 LTV	**23.1**	0.61	0.71	0.67	0.62
*Day 7 \ Day 14 LTV**	28.13	**0.91**	0.85	0.82	0.83
Day 14 \ Day 28 LTV	98.36	0.84	**0.86**	**0.87**	**0.86**

5.4 Offline Inference Test

We conducted an offline inference test to evaluate the performance of the proposed model on real-world data, predicting the LTV of daily incoming users. To emulate a live production environment for the test, we adopted an in-house MLOps system for continuous model updates on a weekly basis.

Table 6: Test result of offline inference on real-world data

Methods	RMSE	R2	WAP	WAR	WAF1
Train Metric	**28.13**	**0.91**	**0.85**	**0.82**	**0.83**
Offline Inference Metric	35.56	0.86	0.76	0.78	0.78

The model achieved the metrics shown in Table 6 for the period from 2023-01-01 to 2023-01-14. The inference data is non-sampled and highly imbalanced, with a ratio of non-payers to payers of about 30:1. Our model showed only 10% difference approximately compared to the training results in Table 3. The reason behind higher WAP and lower WAR than the validation metrics may be due to false positives where the model incorrectly guessed non-payers as payers. It would also decrease the R2 metric compared to the training results. Nevertheless, the overall metrics uphold their validity and demonstrate that the model effectively estimates the LTV.

6 CONCLUSION

In this paper, we propose Multi Datasource LTV User Representation (MDLUR), a universal user embedding for downstream tasks, specifically LTV prediction. The core of our solution lies in an architecture that acknowledges the unique attributes of different data sources. By allocating distinct architectures to each source, we extract the maximum information possible. The aggregation through weighted sum layers and SAE further condenses the outputs from each model. Designed to manage huge features and sparse datasets, MDLUR leverages multiple dimensionality reduction techniques, ensuring efficiency and scalability in the diverse gaming industry applications. These approaches result in a rich and compact understanding of the user, which significantly improves performance in downstream tasks, particularly LTV prediction.

Our experimental results validate MDLUR's strength; it surpasses current industry models in baseline tests and demonstrates compelling performance in offline testing with real-world datasets. Consequently, MDLUR enhances Lifetime Value prediction accuracy, augments the understanding of player behavior and engagement, and amplifies efficiency in the online gaming industry.

Despite these advancements, there is still much room for improvement. Integrating additional data sources to encompass a broader range of aspects could augment the model's performance on various downstream tasks. The exploration of sophisticated architectures and techniques, such as BERT [26] and sampling methodologies [19, 20, 22], could also improve performance. Additionally, we have started to leverage integrated gradients [21] to conduct feature importance testing, which will provide insights into the contribution of each feature on the prediction for each user. This understanding could bring more effective business strategies and personalized player experiences. In the future, we plan to extend the downstream tasks through transfer learning and fine-tuning, while incorporating continual learning [15] for continuously updating the model with new data. Further research will be conducted to fully realize the potential of the MDLUR and improve the performance of the downstream tasks in the gaming industry.

REFERENCES

[1] Paul Bertens, Anna Guitart, Pei Pei Chen, and Africa Perianez. 2018. A Machine-Learning Item Recommendation System for Video Games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)* (Maastricht, Netherlands). IEEE Press, 1–4. https://doi.org/10.1109/CIG.2018.8490456

[2] Pei Pei Chen, Anna Guitart, Ana Fernández del Río, and África Periáñez. 2018. Customer Lifetime Value in Video Games Using Deep Learning and Parametric Models. *2018 IEEE International Conference on Big Data (Big Data)* (2018), 2134–2140.

[3] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior Sequence Transformer for E-Commerce Recommendation in Alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data* (Anchorage, Alaska) *(DLP-KDD '19)*. Association for Computing Machinery, New York, NY, USA, Article 12, 4 pages. https://doi.org/10.1145/3326937.3341261

[4] Si Chen, Yuqiu Qian, Hui Li, and Chen Lin. 2021. Sequential Recommendation in Online Games with Multiple Sequences, Tasks and User Levels. *17th International Symposium on Spatial and Temporal Databases* (2021).

[5] Ana Fernández del Río, Pei Pei Chen, and África Periáñez. 2019. Profiling Players with Engagement Predictions. In *2019 IEEE Conference on Games (CoG)*. 1–4. https://doi.org/10.1109/CIG.2019.8848074

[6] Anders Drachen, Mari Pastor, Aron Liu, Dylan Jack Fontaine, Yuan Chang, Julian Runge, Rafet Sifa, and Diego Klabjan. 2018. To Be or Not to Be...Social: Incorporating Simple Social Features in Mobile Game Customer Lifetime Value Predictions. In *Proceedings of the Australasian Computer Science Week Multiconference* (Brisband, Queensland, Australia) *(ACSW '18)*. Association for Computing Machinery, New York, NY, USA, Article 40, 10 pages. https://doi.org/10.1145/3167918.3167925

[7] Peter S. Fader, Bruce G.S. Hardie, and Ka Lok Lee. 2005. RFM and CLV: Using Iso-Value Curves for Customer Base Analysis. *Journal of Marketing Research* 42, 4 (2005), 415–430. https://doi.org/10.1509/jmkr.2005.42.4.415 arXiv:https://doi.org/10.1509/jmkr.2005.42.4.415

[8] Peter S. Fader, Bruce G. S. Hardie, and Ka Lok Lee. 2005. Implementing the BG/NBD Model for Customer Base Analysis in Excel.

[9] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*.

[10] Google. 2019. Predicting Customer Lifetime Value with AI Platform: Introduction. https://cloud.google.com/architecture/clv-prediction-with-offline-training-intro..

[11] Anna Guitart, Ana Fernández del Río, and África Periáñez. 2019. Understanding Player Engagement and In-Game Purchasing Behavior with Ensemble Learning. *ArXiv* abs/1907.03947 (2019).

[12] Anna Guitart, Shi Hui Tan, Ana Fernández del Río, Pei Pei Chen, and África Periáñez. 2019. From Non-Paying to Premium: Predicting User Conversion in Video Games with Ensemble Learning. In *Proceedings of the 14th International Conference on the Foundations of Digital Games* (San Luis Obispo, California, USA) *(FDG '19)*. Association for Computing Machinery, New York, NY, USA, Article 97, 9 pages. https://doi.org/10.1145/3337722.3341855

[13] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus A. Brubaker. 2019. Time2Vec: Learning a Vector Representation of Time. *CoRR* abs/1907.05321 (2019). arXiv:1907.05321 http://arxiv.org/abs/1907.05321

[14] Kunpeng Li, Guangcui Shao, Naijun Yang, Xiao Fang, and Yang Song. 2022. Billion-User Customer Lifetime Value Prediction: An Industrial-Scale Solution from Kuaishou. In *Proceedings of the 31st ACM International Conference on Information amp; Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 3243–3251. https://doi.org/10.1145/3511808.3557152

[15] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient Episodic Memory for Continual Learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6470–6479.

[16] David C. Schmittlein, Donald G. Morrison, and Richard Colombo. 1987. Counting Your Customers: Who-Are They and What Will They Do Next? *Management Science* 33, 1 (Jan 1987), 1–24. https://doi.org/10.1287/mnsc.33.1.1

[17] David C. Schmittlein, Donald G. Morrison, and Richard Colombo. 1987. Counting Your Customers: Who-Are They and What Will They Do Next? *Manage. Sci.* 33, 1 (jan 1987), 1–24.

[18] Kyuyong Shin, Hanock Kwak, KyungHyun Kim, Minkyu Kim, Young-Jin Park, Jisu Jeong, and Seungjae Jung. 2021. One4all User Representation for Recommender Systems in E-commerce. *ArXiv* abs/2106.00573 (2021).

[19] Rafet Sifa, Fabian Hadiji, Julian Runge, Anders Drachen, Kristian Kersting, and Christian Bauckhage. 2015. Predicting Purchase Decisions in Mobile Free-to-Play Games.

[20] Rafet Sifa, Julian Runge, Christian Bauckhage, and Daniel Klapper. 2018. Customer Lifetime Value Prediction in Non-Contractual Freemium Settings: Chasing High-Value Users Using Deep Neural Networks and SMOTE. In *Hawaii International Conference on System Sciences*.

[21] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

[22] Ali Vanderveld, Addhyan Pandey, Angela Han, and Rajesh Parekh. 2016. An Engagement-Based Customer Lifetime Value System for E-Commerce. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 293–302. https://doi.org/10.1145/2939672.2939693

[23] Udo Wagner and Daniel Hoppe. 2008. Erratum on the MBG/NBD Model. *International Journal of Research in Marketing* 25, 3 (2008), 225–226. https://doi.org/10.1016/j.ijresmar.2008.06.003

[24] Xiaojing Wang, Tianqi Liu, and Jingang Miao. 2019. A Deep Probabilistic Model for Customer Lifetime Value Prediction. *arXiv: Applications* (2019).

[25] Rita D. Wheat and Donald G. Morrison. 1990. Estimating Purchase Regularity with Two Interpurchase Times. *Journal of Marketing Research* 27, 1 (Feb 1990), 87. https://doi.org/10.2307/3172554

[26] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2022. UserBERT: Pre-Training User Model with Contrastive Self-Supervision. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2087–2092. https://doi.org/10.1145/3477495.3531810

[27] Hanting Xie, Sam Devlin, Daniel Kudenko, and Peter Cowling. 2015. Predicting player disengagement and first purchase with event-frequency based data representation. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*. 230–237. https://doi.org/10.1109/CIG.2015.7317919

[28] Mingzhe Xing, Shuqing Bian, Wayne Xin Zhao, Zhen Xiao, Xinji Luo, Cunxiang Yin, Jing Cai, and Yancheng He. 2021. Learning Reliable User Representations from Volatile and Sparse Data to Accurately Predict Customer Lifetime Value. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 3806–3816. https://doi.org/10.1145/3447548.3467079

[29] Bei Yang, Ke Liu, Xiaoxiao Xu, Renjun Xu, Hong Liu, and Huan Xu. 2021. Learning Universal User Representations via Self-Supervised Lifelong Behaviors Modeling. *ArXiv* abs/2110.11337 (2021).

[30] Liangwei Yang, Zhiwei Liu, Yu Wang, Chen Wang, Ziwei Fan, and Philip S. Yu. 2022. Large-Scale Personalized Video Game Recommendation via Social-Aware Contextualized Graph Neural Network. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 3376–3386. https://doi.org/10.1145/3485447.3512273

[31] Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon Jose, Beibei Kong, and Yudong Li. 2021. One Person, One Model, One World: Learning Continual User Representation without Forgetting. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 696–705. https://doi.org/10.1145/3404835.3462884

[32] Yin Zhang, Yujie Li, Ranran Wang, Jianmin Lu, Xiao Ma, and Meikang Qiu. 2020. PSAC: Proactive Sequence-Aware Content Caching via Deep Learning at the Network Edge. *IEEE Transactions on Network Science and Engineering* 7 (2020), 2145–2154.

[33] Shiwei Zhao, Runze Wu, Jianrong Tao, Manhu Qu, Hao Li, and Changjie Fan. 2020. Multi-source Data Multi-task Learning for Profiling Players in Online Games. In *2020 IEEE Conference on Games (CoG)*. 104–111. https://doi.org/10.1109/CoG47356.2020.9231585

[34] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1059–1068. https://doi.org/10.1145/3219819.3219823