A Comprehensive Review of LSTM-Based Churn Prediction Models in the Gaming Industry

Fatemeh Vatani[1], Morteza Dorrigiv*[1], and Samaneh Emami[1]

Abstract**-- Client churn is a significant issue affecting companies across various industries. In the gaming sector, customer loss is particularly critical as it directly impacts revenue, profit margins, and customer retention. Inaccurate predictions of client churn can lead to substantial revenue losses. Churn prediction involves identifying customers who are most likely to cancel their subscriptions. This practice has become essential for many modern organizations due to its performance benefits, aiding businesses in calculating revenue growth and client retention metrics. This paper classifies player churn prediction models into seven main categories to comprehensively review the existing literature. This classification enhances the understanding of various methodologies used in the field and highlights potential areas for future research. Notably, Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, have demonstrated significant potential among deep learning models. This paper examines the contribution of LSTM networks in predicting churn in computer games.**

Index Terms**-- Customer churn, Classification, Deep neural network, Long short-term memory, Survival analysis, Game data mining**

I. Introduction

ONE particular kind of neural network that can process and represent sequential data, including text, audio, time series, and other data types, is the recurrent neural network. Recurrent neural networks rely on both the network's past state and its current input to determine the output of the network at a given period. Because of an issue called vanishing or exploding gradients, these networks do poorly at identifying long-term relationships in the data. One kind of recurrent neural network that can solve the issue of vanishing or exploding gradients is the long short-term memory network. As a result, these networks are very good at processing time series and other sequential data. Long Short-Term Recurrent neural networks, such as memory networks, can solve the issue of disappearing or exploding gradients. As a result, these networks are very good at processing time series and other sequential data. Gates are the building blocks of long short-term memory networks; they are like the memory cells of a computer. Gates manage the information flow across time and keep past data about their condition. Long short-term memory networks have four different kinds of gates. Forget gates determine which historical data to keep and which to dispose of depending on the circumstances. The network can regulate the input in its current state thanks to input gates. A cell state vector is created by adding the contents of the input and forgetting gates. Stated differently, the input gate provides the current state information and the forgetting gate provides the previous state information for the cell. Ultimately, the output gate retrieves the output for the current time slot from the networks. Text, audio, and time series may be processed efficiently by long short-term memory networks because of their special architecture and long-term use of a back-propagation method [1].

Churn is the term for a customer's departure from a service provider. Online games where users can end sessions at any time illustrate this idea [2]. Consumers are among a business's most valuable assets and are crucial to enhancing both performance and market competitiveness. Even in highly competitive markets, consumers have little trouble selecting among a variety of offerings from suppliers. Research indicates that developing a new client is frequently more expensive than maintaining an existing one. Long-term positive relationships with clients mean that a business will make more money from its current clientele [3]. To facilitate management action, gaming firms are searching for methods to identify and anticipate turnover. For this, the game's collected information may be examined [2].

The time between the date of the user's last activity as reported in the data and the anticipated date of his final activity that hasn't been recorded is known as survival time [4]. Survival and churn prediction have been the subject of several research. The models associated with the long short-term memory network are investigated in this study. The remaining portions of the article are structured as follows: In the second section, we discuss the categorization of churn prediction models. The subsequent section reviews studies that have integrated the short-term memory network into their models. The fourth section outlines the performance evaluation criteria, and the conclusion is presented in the fifth section.

II. Categorization of Customer Churn Prediction Models

The modeling of game players' churn prediction has been the subject of much research. User churn has been predicted using

1. Faculty of Electrical and Computer Engineering (ECE), Semnan University, Semnan, Iran.

Corresponding author Email: dorrigiv@semnan.ac.ir

a variety of deep learning and machine learning methods, including feedforward neural networks, random forests, logistic regression, and neural networks. To improve prediction models' accuracy, multiple researchers have proposed new characteristics. Additionally, survival has been examined in research using traditional Cox regression, logistic regression, conditional inference tree approaches, etc. Another team of researchers built a model that took into account both churn analysis and survival factors.

A. Machine Learning Techniques

In [5] research, a two-stage approach was employed to address user churn in mobile games. A churn prediction model was developed utilizing standard classification algorithms, including Logistic Regression, Decision Tree, Random Forest, Naive Bayes, and Gradient Boosting. Subsequently, a targeted notification strategy was implemented to re-engage at-risk users. User behavior was analyzed to identify key game features, and personalized push notifications were delivered to encourage continued gameplay. Four methods were utilized by Rottmayer et al. [2]: Naive Approach, sliding window, quartile, and tendency of several dates. They used accepted definitions of churn from the literature that they found through game analysis to detect churn. The term "churn" often refers to a consumer leaving a service provider. Depending on the term being utilized, churn can be detected. As a result, various labeling strategies have various churn outcomes. For classification, eight machine learning methods were used: neural networks, logistic regression, decision trees, random forests, gradient boosting trees, support vector machines, Naive Bayes classifiers, and k closest neighbors. To forecast survival and churn, Li et al. [6] examined the models employed in the game data mining competition. The participants employed machine learning approaches such as decision trees, LightGBM gradient boosting, logistic regression, random forests, gradient boosting, and extremely random trees in the churn prediction segment.

Wang [7] conducted a study to predict churn among high-value players in free-to-play mobile games. She employed a variety of machine learning algorithms, including linear regression, Naive Bayes, K-Nearest Neighbors, decision trees, random forests, neural networks, and support vector machines. High-value players were defined as those contributing to the top 50% of revenue in the preceding 90 days. This study [8] extracted key features, including player activity metrics and demographics, from the dataset to predict subscription renewal. These features were used to build classification models, such as Logistic Regression, Decision Trees, and others, to identify factors influencing player churn.

David et al. [9] conducted a study to evaluate the performance of Logistic Regression, Random Forest, Support Vector Machines, and Ensemble Learning models for player churn prediction using a publicly available API dataset. This research differs from previous studies that commonly rely on private game logs. Guo et al. [10] utilized a comprehensive set of user features, including static, dynamic, and social network data extracted from game logs, to develop a stack-based model. They employed a stacking ensemble approach that integrated multiple machine-learning algorithms such as Random Forest, LightGBM, XGBoost, CatBoost, ExtraTrees, and K-Nearest Neighbors. The model was structured with a two-layer architecture, where the first layer processed input features independently. The outputs from the first layer were then combined with the original features to create a new feature set, which was fed into the second layer of models. Finally, a voting mechanism combined the predictions from the second layer to produce the final output.

B. Deep Learning Model

To forecast churn, Liu et al. [6] employed a deep neural net. To address micro-level prediction, they put forth a novel semi-supervised and inductive embedding model that simultaneously learns the embedding and the prediction function of the user-application interaction. Deep neural networks, thanks to a novel edge embedding approach, can simulate these two functions and capture connection dynamics and contextual information. Churn prediction was achieved by Zheng et al. [11] using a fully connected feedforward neural net. In order to simulate the user's behavioral patterns, they suggested using an end-to-end neural network and taking into account both the user's in-game and login behaviors.

Glimchi et al. [12] presented a churn prediction model based on sentiment analysis in mobile games utilizing word embedding models and deep learning techniques. Word embedding models were employed for text display and deep learning methods were used for categorization. The models that were employed in the game data mining competition to forecast survival and churn were discussed in Lee et al.'s paper [4]. The feed-forward neural network, deep neural network, and long short-term memory combined are the deep learning models that were employed in the churn prediction section. The proposed model [13] uses a blend of machine learning techniques to predict user churn, integrating sequential pattern mining, Graph Convolutional Networks (GCNs), and Long Short-Term Memory (LSTM) networks to analyze user behavior patterns. It also introduces a novel gating mechanism to tackle data sparsity issues.

C. Explainable AI Method

Xiong et al. [14] propose a comprehensive, data-driven framework to tackle player churn in online games. Drawing on domain expertise, the study employs and enhances state-of-the-art eXplainable AI (XAI) techniques. LightGBM is used for churn prediction, while TreeSHAP and an optimized Anchor method are employed for model interpretability. The optimized Anchor method is further refined through intermediate variable reuse, parallel processing, and advanced discretization. This study [15] proposes a framework, GXAI, to analyze player churn in online games. It combines various machine learning models (e.g., Random Forest, LightGBM, CNN, LSTM, GCN, Transformer) with explainability techniques (e.g., TreeSHAP, DeepExplainer) to understand player behavior and predict churn. The framework also includes methods for generating different types of explanations (individual, local, and global) to support decision-making for game operators and designers.

D. Prediction Models Using New Features

This study [16] extracted both raw and cognitive psychological features from game logs. Raw features encompassed login, character, battle, and purchase data, while cognitive features captured engagement, activity, and personality traits. These features were employed to predict churn using decision trees and random forest models. Kim et al.[17] focused on newly acquired players, formalizing a churn definition based on observation and prediction periods. Subsequently, a standardized churn analysis process tailored for casual games was developed. This process encompassed data preprocessing, feature engineering, and churn prediction modeling. Traditional machine learning algorithms, including logistic regression, gradient boosting, and random forests, were employed alongside deep learning approaches like CNN and LSTM.

A commitment-based machine learning strategy was given by Martínez-Comer et al. [18] to address two risky scenarios that arise during the game usage cycle: churn prediction and player lifetime remaining. To allocate a player's contribution to a game based on facts, they put up the notion of commitment. Determined commitment metrics and established a propensity degree for each. The idea behind adding new features is to use observed changes to offer hints about potential future behavior. In order to determine the desired behaviors of the user, it is intended to assess all player engagement activities and then develop new features based on the results. Using various neural network architectures, Christensen et al. [19] looked at ways to combine ordinal and cumulative data to enhance the state of churn prediction.

Using logistic regression analysis, Perisic et al. [20] concentrated on developing a churn prediction model for the mobile games industry within the context of purchase Recency, Frequency, and Monetary (RFM). The feature's importance was assessed using dominance analysis and Robust statistical measures. Regardless of the users' lifespan, a logistic model has been developed using the collected attributes to forecast churn and categorize potential defectors within a user community. To forecast churn in the context of mobile games, the researchers in their follow-up study [21] integrated the characteristics linked to user Lifespan, Intensity, and Reward (LIR) with the data presented in their earlier study and a feature framework (RFM-LIR).

The creation of generic features to forecast churn for long-term participants was examined by Yang et al. [22]. More specifically, they took two free-to-play online games and retrieved the order in which players spent their time in each game. To forecast churn, they employed generic characteristics that contained information on the sequence in which players spent their time in the game. In their subsequent study [23], they explored the creation of novel generic characteristics that take into account players' playing time in order to forecast churn for loyal gamers. They used the information theory concepts of entropy and cross-entropy to assess the order of game time in order to achieve this goal. Next, they predicted the churn using generic characteristics derived from the participants' playing time order information.

Óskarsdóttir et al.[24] investigated the influence of social networks on player retention within a mobile game. To accomplish this, they constructed two types of networks: a friend network based on explicit social connections and a similarity network based on player attributes. By analyzing these networks, they extracted features related to player interactions and churn behavior. These features were then integrated into predictive models to enhance the accuracy of forecasting player churn. This research [25] proposes a novel approach to predict player churn in MMORPGs by leveraging Graph Convolutional Networks (GCNs) to model in-game social interactions. By focusing on party, guild, and item trading relationships, the study extracts valuable features to improve churn prediction accuracy. To further enhance the model, the Correct and Smooth (C&S) method is incorporated to refine predictions.

E. Survival Prediction Models

A survival ensemble model was developed by Perianez et al. [26] to provide comprehensive analysis and accurate prediction of user churn. By combining multiple survival trees, this approach offers granular insights into player behavior and churn likelihood over time. The model leverages conditional inference survival trees, incorporating weighted Kaplan-Meier estimates and linear rank statistics for enhanced performance. In their next research [27], they employed a two-step process to optimize the model. First, the optimal split variable was selected based on covariate-response relationships, followed by determining the optimal split point through a comparison of two-sample linear statistics across all potential partitions. Subsequently, a parallelized version was implemented to enhance efficiency and expand the model's applicability to additional response variables such as level and playtime.

The models utilized by the teams competing in the data mining game competition in the survival analysis component of the study by Lee et al. [4] comprised the generalized linear model, logistic regression, gradient boosting, conditional inference tree, and collective class tree. To anticipate player churn, Demdiuk et al. [28] proposed a solution to the survival analysis problem using Cox regression with mixed effects. Initially, they modeled the survival function at the population level using the Kaplan–Meier estimator. Subsequently, they employed both mixed effects and conventional Cox regression models to examine the impact of behavioral factors.

F. Models for Simultaneously Predicting Churn and Survival

A user interaction model that models the first user-game interactions and combines lengthy short-term memory and neural networks is described by Bonometti et al. [29]. This model enables combined calculation of survival time and churn probability. By introducing the churn vector, Zhang et al. [30]developed a method to take into account each user's unique usage duration in churn prediction. They employed Lasso, gradient boosting, random forest, decision tree, neural network, and support vector machine algorithms for regression and classification, with an emphasis on user churn prediction and churn time (survival). In their subsequent work [31], the researchers employed a variety of techniques for regression and classification, including Lasso, support vector machines, random forests, gradient boosting, multilayer perceptron, convolutional neural networks, recurrent neural networks, long short-term memory, and attention networks.

G. Multi-objective approach

Roohi et al. [32] introduced a novel approach to predicting player churn and pass rates in free-to-play games by integrating deep reinforcement learning (DRL) with a population-level simulation. By training an AI agent using DRL to play the game, they obtained accurate estimates of level difficulty, which were then used to drive a simulation of player behavior. This combined approach enabled a more nuanced understanding of player dynamics and churn patterns. In their next work [33], they have developed a model that addresses player churn in free-to-play games by combining AI and player behavior analysis. The model leverages Deep Reinforcement Learning to estimate the level of difficulty and simulate player behavior. To personalize the gaming experience, they calculate Perceived Difficulty (PPD), measuring the gap between expected and actual player effort. Additionally, they assess Dynamic Difficulty Influence (DDI) to understand how a player's sensitivity to difficulty changes over time. To quantify DDI, they utilize a time-dependent survival analysis model known as D-Cox-Time. By incorporating PPD and DDI, the model predicts churn, enabling real-time difficulty adjustments to enhance player retention and engagement.

del Río et al. [34] categorized free-to-play game players into three segments: non-paying active users, paying active users, and inactive users. A state-space time series model, incorporating ARIMA and unobserved components, was employed to analyze daily conversion rates between these segments and churn prediction. Karmakar et al. [35] collected data on player engagement (time spent playing), collaboration (interactions with other players), and achievement (game progress) at each completed level. To control for potential time-of-day effects, especially on collaborative behavior, they included a time-of-day variable in their analysis. They developed a statistical model to predict player churn based on this data. Using a generalized linear mixed model (GLMM), they investigated the relationships among engagement, collaboration, achievement, and churn across different game levels. As detailed in Section C, Tao et al [15] present a model capable of analyzing individual player behavior patterns to detect anomalies indicative of potential cheating.

III. CHURN AND SURVIVAL PREDICTION MODELS USING LONG SHORT-TERM MEMORY

We will go over the current studies on long short-term memory networks in this part.

To anticipate churn, the authors [4] and [36] presented a model that combines a deep neural network with long short-term memory. To analyze the time series data and learn a single vector representation that explains the behavior of the time series, a multi-layer long short-term memory network is first used. Subsequently, the output of a multi-layer deep neural network trained on static characteristics is combined with this vector. To forecast the binary result's ultimate output layer, a second layer was trained on top of the combined representation following the merger. Every layer employed dropout to prevent overfitting. Because there were a lot of correlated characteristics, the model's generalization is enhanced and it avoids being too dependent on a single feature by only choosing a portion of the input time series and static data. In **Error! Reference source not found.**., the replicated model [4] is displayed.

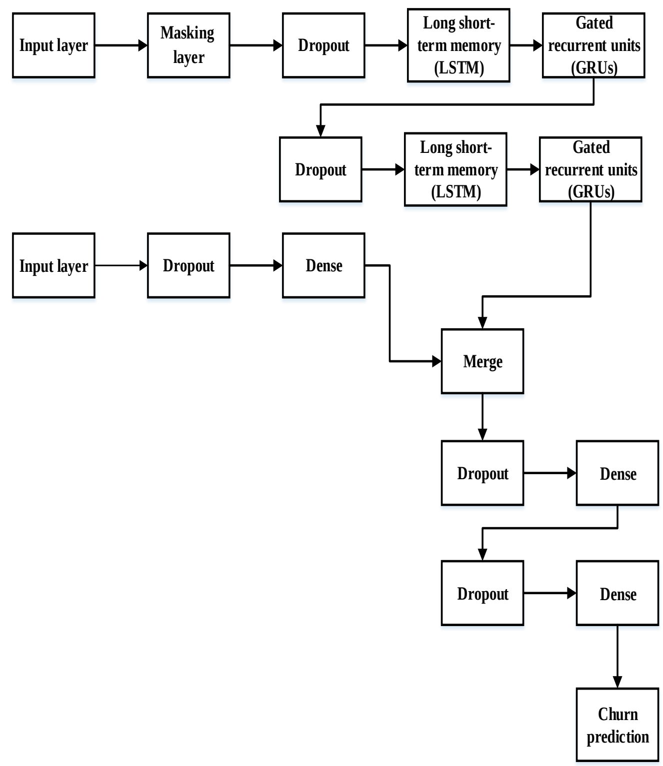

Fig. 1. The combined structure of long short-term memory with a deep neural network (reproduced from [4])

Error! Reference source not found.. displays the architecture designed by the authors [29]. A feature vector and a content vector with the game's numerical encoding are fed into this model. Subsequently, the game text is included in a vector in a manner akin to that of embedded words utilized in emotion analysis. Next, along the temporal dimension, the embedded game content vector and the raw behavioral input are concatenated to create a single, zero-padding feature vector that may be reused as needed. Currently, working with a time series of varying durations and applying a dense layer to each time step is possible thanks to the masking layer. This layer allows the model to create a new content vector by fusing content with raw behavioral characteristics. A long short-term memory is used to model these just-learned characteristics. As a result, a feature vector—a concealed representation of the input characteristics across time and a high-level representation of the user's behavioral state throughout the observation period—is the layer's output to the long short-term memory. After obtaining this high-level latent representation, the architecture's last step transfers it to two shallow neural networks for survival and churn prediction. The survival time and the likelihood of churn are the outputs of these estimators, which are made up of two linked dense layers.

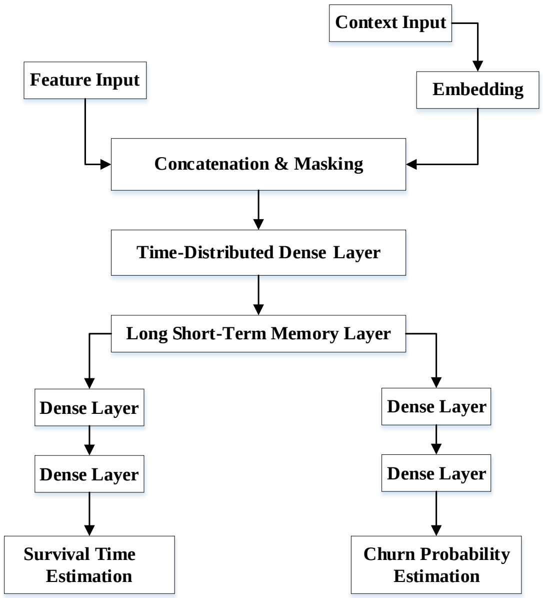

Fig. 2. Bifurcation model (reproduced from [15])

There are three primary parts to the research model [11]: 1) the game incorporates a behavior encoder that represents every user's behavioral data as an embedded vector. 2) A login behavior encoder that represents each user's login details as a vector embedded in the context of the online game. 3) A layer of combination and prediction that adds together more than two different kinds of embedding vectors and produces the ultimate likelihood of the user quitting the game. We will now go over the specifics of the model depicted in Fig. 3.

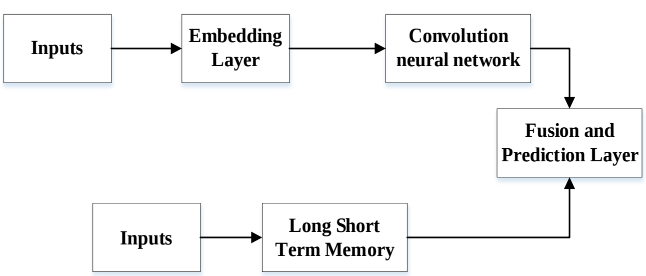

Fig. 3. The architecture of the churn model (reproduced from [6])

The game's behavior encoder:

1) Input: Daily occurrences are gathered for the user and presented in chronological order as in-game behavioral sequences, providing them with in-game information. The extensive previous user behavior may significantly raise the complexity of the model and lengthen the training period. T_1 data from the day before the observation began are the source data utilized in this article.
2) Embedding layer: Using an embedding layer, events are concealed within content vectors in a hidden area. Every event in the discretization process is coded as a vector with only one item. An embedding layer is used to transform the high-dimensional binary vectors that make up the inputs into a dense representation.
3) A time-aware filtering system that takes into account user behavior and its relationship to the day it occurs. To recognize these distinctive actions according to time intervals, a time-aware gating system is introduced. To find out which traits spread to the downstream layers, a time matrix is added for every day.
4) Multiview mechanism: Convolutional units have been utilized to create multiview because convolutional neural network convolutional filters have been successful in collecting local information for image identification. Convolutional neural network units are thought to produce polynomials. The following patterns are viewed as local aspects of the input, which is regarded as an "image" of behavioral data. Convolutional filters take the form of matrices that are rotated across "images" to summarize various combinations of behaviors from various viewpoints. Daily, filters are utilized to encode in-game activities.

Login Behavior Encoder: Online game login behaviors may be described by factors like playing time, frequency of logins, etc. The login details are described on this page using each user's daily game time. T_2 consecutive time windows prior to the observed day are taken into consideration and a time window of size T_1days is defined. The sequence$m_u = \{m_1, m_2, \dots, m_{T_2}\}$, where m_t is a vector$|T_1|$, can be used to indicate the input. The length of each day is displayed in t windows in the next one.

Recurrent neural network layer: Both long-term and short-term modeling have been done using the long short-term memory of several layers. This network was selected because it uses distinct time periods of daily data as time series and is very capable of learning intrinsic temporal correlations to illustrate the range of activity.

To examine the churn forecast based on the analysis of players' emotions for each game, the research [12] consists of three key stages. These phases are, in order, data preprocessing, classification, and data collecting. Using Selenium's crawler, player comments are gathered from Google Play Store sites during the data-gathering stage. Next, each mobile gaming app's rating, associated player username, date, vote, and remark details are gathered. Before beginning the classification phase in the second stage, the raw text dataset is cleaned using preprocessing techniques. The Zemberek library is used to eliminate stop words, punctuation, and emoticons. Comments from users are ranked from 1 to 5. To assess the tone of the comments made by the players, it is determined that comments 1, 2, and 3 are negative while comments 1 through 3 are positive. The next step involves evaluating churn prediction based on user sentiment analysis using deep learning algorithms and word embedding techniques. This study uses Word2Vec, GloVe, and FastText as word embedding techniques and convolutional neural networks, recurrent neural networks, and short-term memory networks as deep learning models.

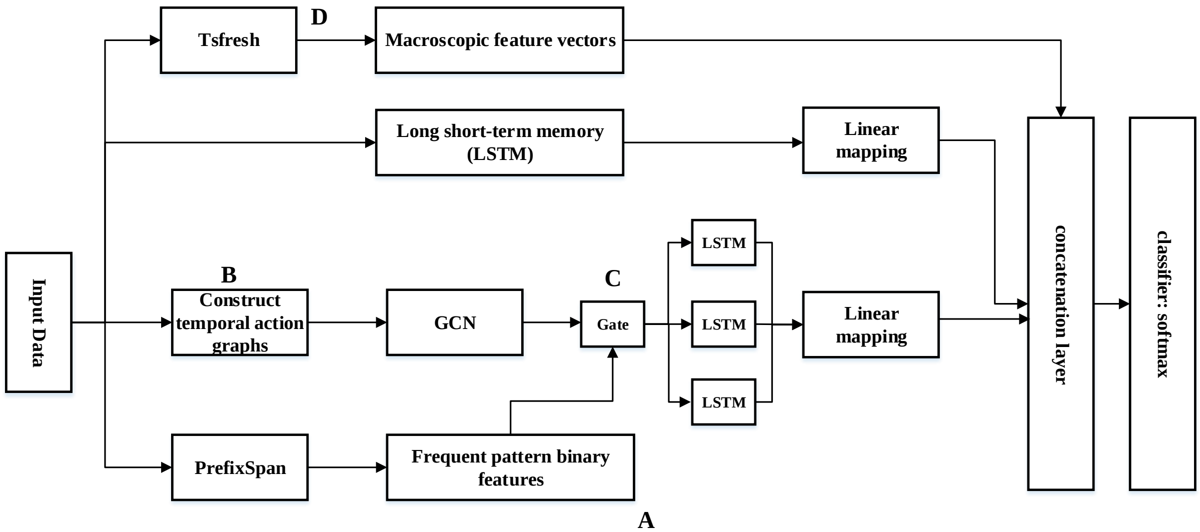

Fig. 4. Overview of the framework Pattern-based Graph with Convolutional Neural Network (PGCNN), featuring pattern matrix expansion (A), time-interval based action graph construction (B), pattern gate (C), and the Tsfresh pre-processing component (D). The linear mapping is achieved through a simple dense layer (reproduced from [13])

While deep learning models handle the classification problem, word embedding models supply word vectors. The output of each word embedding model is fed into multilayer perceptrons in order to capture the influence of the models. Therefore, word embedding techniques also undergo categorization efforts. Sentiment analysis uses word embedding models due to their effectiveness. word vectors in a document that are comparable to other words in a document are provided by word embedding models. By using word embedding models, syntactic and semantic similarity as well as the correlation between terms inside a certain collection context is guaranteed.

Using a short-term memory encoder, the complete input is processed and encoded into a content vector in the paper [30]. To attain superior precision in the churn prediction, a churn vector is established using equation (1):

$$ChurnVector = \frac{RemainDays}{LastPlayDay - FirstPlayDay} \quad (1)$$

Equation (1) illustrates that the churn vector is the normalized number of days left to the total number of days the user has played. The business can effectively advertise to potentially divert users with the aid of this vector. Churn vectors allow you to classify various user groups and target each group with a distinct advertising strategy. Lastly, a variety of models and algorithms, including long short-term memory, have been used to forecast the churn based on the date and the churn vector.

To predict user churn, Halbersberg et al. [13] propose a novel framework that extracts multiple representations of user behavior. This framework incorporates frequent sequential pattern mining, temporal action graphs processed by a GCN, a gating mechanism, an LSTM, and a Tsfresh component [37]. These representations are combined and fed into a softmax classifier to predict churn propensity. Their contributions extend the GCN-LSTM framework by introducing these additional components and a novel gating mechanism.

A) *Frequent pattern matrix:* The proposed model employs a sequential pattern mining algorithm to identify unique patterns in churner and non-churner user behavior. These patterns are then combined to create a joint pattern set, which serves as the foundation for subsequent analysis and classification. Essentially, the model seeks to distinguish between users who churn and those who do not by identifying sequences of actions that are characteristic of each group.

B) *Temporal action node features*: The method involves creating a graph for each user session, where nodes represent actions and edges represent transitions between actions. Importantly, these edges are weighted based on the average time it takes to move from one action to another. To make these time-based comparisons meaningful, the edge weights are normalized based on the average transition times across all users. This ensures that comparisons between different user sessions are fair and accurate. Once the weighted action graph is constructed, a Graph Convolutional Network (GCN) is employed to extract features from it. This neural network is designed to handle graph-structured data and can capture complex relationships between actions and their temporal dependencies.

C) Node selection gate: The sparsity of graphs, due to numerous isolated nodes with few connections, hampers machine learning models' ability to learn meaningful patterns from limited data. The authors propose a node selection mechanism that focuses on a subset of nodes deemed more important based on their relevance to frequent sequential patterns in user behavior. The node selection process is implemented as a gating mechanism within the model, where the GCN component initially processes all nodes but the gate selectively passes information from nodes involved in frequent sequential patterns.

D) Demonstration of user journey processing in the PGCNN: The PGCNN model processes user journeys as graphs, converting them into matrices for a GCN to generate node features. A node selection gate filters important nodes based on frequent sequential patterns, and these features are then processed through an LSTM layer along with traditional sequential and statistical methods before being fed into a softmax layer to predict churn.

IV. Performance evaluation criteria

To assess the models' performance, one must look at their performance and compare their prediction power. Several performance evaluation criteria have been put forth to compare the efficacy of various survival and churn models. Any model that is constructed with both balanced and unbalanced data can have its performance examined using these metrics. The following explains the criteria.

A) Accuracy

Accuracy is a widely applicable performance measure, but in some cases, it is not the best one, particularly when the target variable classes in the dataset are imbalanced. The formula (2) is used to determine a prediction model's accuracy. The numbers that correspond to the correct positives, false positives, false negatives, and correct negatives are indicated by the symbols TP, TN, FN, and FP.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2)$$

B) Recall

This metric indicates the proportion of observations that the algorithm correctly predicts as positive. Recall can be calculated using formula (3), which divides the total number of correct positives by the sum of the correct positives and false negatives.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

C) Precision

This metric indicates the proportion of positive observations that an algorithm actually predicted to be positive. Precision is determined by dividing the total number of true positives by the sum of true positives and false positives, as stated in formula (4):

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

D) F1 score

This measure combines precision and recall. F1 score is defined by equation (5):

$$F1 = \frac{precision \times recall}{precision + recall} \quad (5)$$

E) RMSLE

Formula (6) is used to calculate this criterion. O_i and P_i are the observed and predicted survival times, respectively.

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(o_i + 1) - log(p_i + 1))^2} \quad (6)$$

F) Symmetric mean absolute percentage error

Equation (7) is used to calculate this criterion. The predicted time and the actual value for user i are represented by the variables $\hat{y}_i$ and y_i, respectively, in this formula, where n represents the set of all users.

$$SMAPE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\sum_{i=1}^{n}(\hat{y}_i - y_i)} \quad (7)$$

G) Coefficient of determination

Equation (8) is used to calculate the detection coefficient. The total sum of squares associated with the outcome variable, which is the sum of the squares of the measurements minus their mean, and the sum of the differences between the predicted value and the mean of the dependent variable are represented, respectively, by the parameters TSS and ESS.

$$R^2 = \frac{ESS}{TSS} \quad (8)$$

The criteria used in evaluating the models are listed in **Error! Reference source not found.** I, along with information about the datasets used in the reviewed studies.

Data was collected from 10,000 Blade & Soul players, totaling 100GB of game logs. This log data included detailed information about player actions, items, and interactions within the game. Predictions were made three weeks in advance, with churn defined as no activity within a subsequent five-week period. The winning team of [4] extracted over 3,000 features from the player data, encompassing daily actions, sessions, playtime, levels, equipment scores, and financial information. These features were further processed using various statistical methods to generate additional features. Feature selection techniques, including LSTM Autoencoder and Feature Value Distribution, were employed to refine the feature set. The study [29] utilized data from six Square Enix games to analyze user churn. This dataset, encompassing various game genres and platforms, provided a diverse foundation for the research. Data cleaning focused on removing outliers to ensure reliability. The researchers defined five key metrics to measure player behavior within game sessions: session time, total playtime, time between sessions, activity level, and activity diversity, while also considering the game context. The dataset was split into training, validation, and test sets for model development and evaluation, with feature scaling applied to normalize the data. The researchers in [11] extracted daily user events and arranged them chronologically to form behavioral sequences. The length of each sequence, representing daily playtime, was normalized.

The dataset was then split into training, validation, and testing sets, with downsampling applied to address class imbalance.

TABLE I
The Evaluation Criteria of the Reviewed Researches and the Used Datasets

Evaluation criteria	Date/duration of data collection	Number of Churners	Total number of players	Dataset		Article
				Game	company	
F1-Score RMSLE	Apr-1-2016 – May-11-2016 July-27-2016 – Sep – 21-2016 Dec-14-2016 – Feb-11-2016	3000	10000	Blade & Soul	NCSOFT	[4]
F1-Score SMAPE	From the release of the game until February 2019	480000	480000	Hitman Go, Hitman Sniper, Just Cause, Just Cause, Life is Strange, and Life is Strange: Before the Storm	Square Enix Limited	[29]
F1-Score Precision Accuracy	June-22-2018 – Sep-20-2018	354561	1063683	-	NetEase	[11]
Accuracy	-	-	42044	Hitman, Soccer Manager, Grand Theft Auto: San Andreas, City Driving 3D	-	[12]
R^2 score Accuracy	8 months	-	-	MAZE X BRAVE	-	[30]
F1-Score Precision Recall Accuracy	Nov-1-2021– Nov-15-2021	375000	1250000	-	Playtika	[13]

Study [12] collected user comments from the Google Play Store for four mobile games. The data was gathered using a Selenium crawler and preprocessed to remove noise, such as emojis, punctuation, and stop words. Sentiment analysis was then performed by classifying the comments as positive or negative based on a 1 to 5 rating scale. The authors of [30] collected data on player behavior, such as logins, game progress, resource acquisition, and character development, to analyze user patterns and predict retention rates during the last five months of the study. They extracted various factors from user play logs to understand behavior and predict churn, categorizing these factors into actions, records, changes, and achievement-related metrics. To enhance their prediction model, they used Bayesian optimization to efficiently tune the hyperparameters.

The research [13] dataset consists of 150 million actions recorded within the first 48 hours of installation, including timestamps and user IDs. To reduce computational complexity, the dataset was narrowed down to the top 5% of sessions with up to 100 actions. Sessions with fewer than three actions and users with fewer than three sessions were excluded. Handling missing values in the action graphs was a key challenge. The authors experimented with various methods, including substituting missing values with -1, the average value across all users, or a large value representing infinite time. Replacing missing values with -1 provided the best results.

With an F1 score of 0.062—the highest value among the other examined models—model [4]outperformed the other 13 models in the churn prediction section. The research's bifurcation model [29]also reported a symmetric mean absolute error percentage of 0.26 and an average F score of 0.168 across six sets. The real value, prediction error, and percentage of absolute error are all displayed by the symmetric average; the higher the number for the F1 scores, the lower the value. When compared against both independent and integrated churn prediction and classification algorithms, this model performs the best.

The best average for this model in three data sets is 0.51 for precision, 0.60 for the F1 score, and 0.67 for accuracy, according to comparative results between the churn model [6] and advanced models. This is because online gaming companies take into account factors like login credentials and in-game activity data. To identify possible patterns of behavior for in-game actions, it employs a multi-view technique. It is evident that the model captures dynamic elements in the log data since it is sensitive to fluctuations in the daily gaming time. The model is upgraded significantly by combining the two kinds of data.

Using several data sets of Turkish mobile gaming programs, the impact of deep learning models and word embedding techniques [12]was assessed in terms of various training set percentages, including 80, 50, and 30. With an accuracy of

81/52, 81/37, and 80/50, respectively, the long short-term memory model is used in various percentages of the training set, such as 80, 50, and 30. Long short-term memory does poorly in this comparison. Regression and classification have been performed using the research model [30]. The majority of algorithms that used the churn vector had higher scores.

The experiments demonstrate that PGCNN [13] outperforms the base models and other approaches, achieving an F1-score of 0.61, Precision of 0.71, Recall of 0.53, and Accuracy of 0.81. In summary, the integration of temporal information, common patterns, and a combination of various techniques makes the PGCNN model significantly more effective in predicting user abandonment compared to other models.

As can be observed, one of the difficulties this study is confronting is the lack of a universal standard for comparing and evaluating various outcomes. Researchers have put forth a variety of data mining techniques to effectively handle the churn prediction problem. It takes an effective forecasting model to prevent the churn issue. To forecast churn, a good prediction model can handle big inputs with lots of dimensions and intricate attributes.

Artificial intelligence is increasingly being leveraged for churn prediction, with machine learning techniques at the forefront of enhancing predictive accuracy and uncovering deeper insights into customer churn patterns. The industry is witnessing a paradigm shift towards advanced analytics, coupled with a strong emphasis on retention strategies and the integration of big data and AI. These technological advancements are reshaping the game industry's approach to customer retention and churn management.

A significant proportion of existing research relies solely on a single data mining method for churn prediction, potentially neglecting the advantages of ensemble techniques in enhancing predictive accuracy and providing a deeper understanding of customer behavior. While efforts have been made to refine predictive models, a comprehensive exploration of the root causes driving customer churn remains limited. Given the pivotal role of these factors in developing effective retention strategies, further investigation is warranted.

Despite the growing importance of interpretability in machine learning, limited research has been conducted on applying XAI methods to the domain of churn prediction. Future studies should address this gap by developing and evaluating XAI-based models for churn analysis. While the gaming industry has been a focal point, the applicability of churn models to other sectors, such as banking, energy, and social media, remains underexplored. The knowledge and models developed within the gaming industry could potentially be adapted for application in other sectors. A customer-centric approach, involving segmentation based on demographic and behavioral factors, offers a promising avenue for enhancing churn prediction and retention efforts.

Acquiring datasets from the gaming industry for churn prediction presents significant challenges. Many companies are reluctant to share their data due to competitive concerns, privacy issues, and the proprietary nature of their data. This scarcity of publicly available datasets hinders research in this area.

V. Conclusion

To address the issue of churn customers in the gaming business, this study first looked into the topic of churn prediction and the significance of applying predictive modeling techniques. We thoroughly examined the churn prediction techniques currently in use that made use of long short-term memory networks. This paper provided the datasets utilized in these investigations, in contrast to other review publications that mostly concentrated on prediction models and the accuracy of churn prediction. Lastly, we looked at a set of criteria that have been employed in studies to assess how well various churn-predicting techniques work.

By examining player data, game creators may spot churners and keep players interested by giving them discounts on subscription fees, in-game purchases, or other promotional offers. As previously indicated, retaining current customers is far less expensive than bringing in new ones. Therefore, accurate prediction can significantly save game makers' expenses. Researchers may gain a better understanding of churn prediction, models and methodologies, performance indicators, and gaming data sets that are utilized to apply these strategies by reading this research.

VI. References

[1] CU. Singh, S. Murugesan, and A. Seth, Emerging Computing Paradigms: Principles, Advances and Applications. Wiley-Blackwell, 2022.

[2] K. Rothmeier, N. Pflanzl, J. A. Hullmann, and M. Preuss, “Prediction of Player Churn and Disengagement Based on User Activity Data of a Freemium Online Strategy Game,” IEEE Trans. Games, vol. 13, no. 1, pp. 78–88, Mar. 2021, doi: 10.1109/TG.2020.2992282.

[3] X. Xiahou and Y. Harada, “B2C E-Commerce Customer Churn Prediction Based on K-Means and SVM,” J. Theor. Appl. Electron. Commer. Res., vol. 17, no. 2, pp. 458–475, Apr. 2022, doi: 10.3390/jtaer17020024.

[4] E. Lee et al., “Game Data Mining Competition on Churn Prediction and Survival Analysis Using Commercial Game Log Data,” IEEE Trans. Games, vol. 11, no. 3, pp. 215–226, Sep. 2019, doi: 10.1109/TG.2018.2888863.

[5] M. Milošević, N. Živić, and I. Andjelković, “Early churn prediction with personalized targeting in mobile social games,” Expert Syst. Appl., vol. 83, pp. 326–332, 2017, doi: 10.1016/j.eswa.2017.04.056.

[6] X. Liu et al., “A Semi-Supervised and Inductive Embedding Model for Churn Prediction of Large-Scale Mobile Games,” in 2018 IEEE International Conference on Data Mining (ICDM), IEEE, Nov. 2018, pp. 277–286. doi: 10.1109/ICDM.2018.00043.

[7] G. Y. Wang, “Churn Prediction for High-Value Players in Freemium Mobile Games: Using Random Under-Sampling,” Statistika, vol. 102, no. 4, pp. 443–453, 2022, doi: 10.54694/STAT.2022.18.

[8] Y. Hossain, E. Azizi, and L. Zaman, “Predicting subscription renewal using binary classification in World of Warcraft,” Entertain. Comput., vol. 44, no. March 2022, p. 100522, 2025, doi: 10.1016/j.entcom.2022.100522.

[9] D. David and A. Zahra, “Player Churn Prediction In Free To Play Game Using Ensemble Learning,” Action Res. Lit., vol. 8, no. 4, pp. 543–549, 2024, doi: 10.46799/arl.v8i4.280.

[10] R. Guo, W. Xiong, Y. Zhang, and Y. Hu, “Enhancing Game Customer Churn Prediction with a Stacked Ensemble Learning Model,” pp. 0–26, 2024.

[11] A. Zheng, L. Chen, F. Xie, J. Tao, C. Fan, and Z. Zheng, “Keep You from Leaving: Churn Prediction in Online Games,” 2020, pp. 263–279. doi: 10.1007/978-3-030-59416-9_16.

[12] Z. H. Kilimci, H. Yoruk, and S. Akyokus, “Sentiment Analysis Based Churn Prediction in Mobile Games using Word Embedding Models and Deep Learning Algorithms,” in 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), IEEE, Aug. 2020, pp. 1–7. doi: 10.1109/INISTA49547.2020.9194624.

[13] D. Halbersberg, R. Yasinnik, M. Halevi, and M. Salhov, “Temporal-Based Action Graph with Sequential Pattern Mining for Churn Detection : a Playtika Case Study,” 2023 Int. Conf. Mach. Learn. Appl., pp. 324–331, 2023, doi: 10.1109/ICMLA58977.2023.00052.

[14] Y. Xiong et al., “A Data-Driven Decision Support Framework for Player Churn Analysis in Online Games,” Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 5303–5314, 2023, doi: 10.1145/3580305.3599759.

[15] J. Tao et al., “Explainable AI for Cheating Detection and Churn Prediction in Online Games,” IEEE Trans. Games, vol. 15, no. 2, pp. 242–251, 2023, doi: 10.1109/TG.2022.3173399.

[16] J. Jeon, D. Yoon, S. Yang, and K. Kim, “Extracting gamers’ cognitive psychological features and improving performance of churn prediction from mobile games,” 2017 IEEE Conf. Comput. Intell. Games, CIG 2017, pp. 150–153, 2017, doi: 10.1109/CIG.2017.8080428.

[17] S. Kim, D. Choi, E. Lee, and W. Rhee, “Churn prediction of mobile and online casual games using play log data,” PLoS One, vol. 12, no. 7, pp. 1–19, 2017, doi: 10.1371/journal.pone.0180735.

[18] L. B. Martins Kummer, J. Cesar Nievola, and E. C. Paraiso, “Applying Commitment to Churn and Remaining Players Lifetime Prediction,” in 2018 IEEE Conference on Computational Intelligence and Games (CIG), IEEE, Aug. 2018, pp. 1–8. doi: 10.1109/CIG.2018.8490443.

[19] J. T. Kristensen and P. Burelli, “Combining Sequential and Aggregated Data for Churn Prediction in Casual Freemium Games,” in 2019 IEEE Conference on Games (CoG), IEEE, Aug. 2019, pp. 1–8. doi: 10.1109/CIG.2019.8848106.

[20] A. Perisic and M. Pahor, “Extended RFM logit model for churn prediction in the mobile gaming market,” Croat. Oper. Res. Rev., vol. 11, no. 2, pp. 249–261, 2020, doi: 10.17535/crorr.2020.0020.

[21] A. Perisic 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, doi: 10.1109/TG.2021.3067114.

[22] W. Yang et al., “Mining Player In-game Time Spending Regularity for Churn Prediction in Free Online Games,” in 2019 IEEE Conference on Games (CoG), IEEE, Aug. 2019, pp. 1–8. doi: 10.1109/CIG.2019.8848033.

[23] W. Yang, T. Huang, J. Zeng, L. Chen, S. Mishra, and Y. E. Liu, “Utilizing Players’ Playtime Records for Churn Prediction: Mining Playtime Regularity,” IEEE Trans. Games, vol. 14, no. 2, pp. 153–160, Jun. 2022, doi: 10.1109/TG.2020.3024829.

[24] M. Óskarsdóttir, K. E. Gísladóttir, R. Stefánsson, D. Aleman, and C. Sarraute, “Social networks for enhanced player churn prediction in mobile free-to-play games,” Appl. Netw. Sci., vol. 7, no. 1, 2022, doi: 10.1007/s41109-022-00524-5.

[25] Y. Han and J. Moon, “Prediction of Churning Game Users Based on Social Activity and Churn Graph Neural Networks,” IEEE Access, vol. 12, no. June, pp. 101971–101984, 2024, doi: 10.1109/ACCESS.2024.3429559.

[26] A. Perianez, A. Saas, A. Guitart, and C. Magne, “Churn Prediction in Mobile Social Games : Towards a Complete Assessment Using Survival Ensembles”.

[27] P. Bertens and A. Guitart, “Games and Big Data : A Scalable Multi-Dimensional Churn Prediction Model”.

[28] S. Demediuk et al., “Player retention in League of Legends,” in Proceedings of the Australasian Computer Science Week Multiconference, New York, NY, USA: ACM, Jan. 2018, pp. 1–9. doi: 10.1145/3167918.3167937.

[29] V. Bonometti, C. Ringer, M. Hall, A. R. Wade, and A. Drachen, “Modelling Early User-Game Interactions for Joint Estimation of Survival Time and Churn Probability,” in 2019 IEEE Conference on Games (CoG), IEEE, Aug. 2019, pp. 1–8. doi: 10.1109/CIG.2019.8848038.

[30] K. Jang, J. Kim, and B. Yu, “On Analyzing Churn Prediction in Mobile Games,” in 2021 6th International Conference on Machine Learning Technologies, New York, NY, USA: ACM, Apr. 2021, pp. 20–25. doi: 10.1145/3468891.3468895.

[31] K. Jang, J. Kim, and B. Yu, “Vector-based Churn Prediction using Neural Networks in Mobile Games,” in 2019 IEEE International Conference on Big Data (Big Data), IEEE, Dec. 2019, pp. 6081–6083. doi: 10.1109/BigData47090.2019.9006052.

[32] S. Roohi, A. Relas, J. Takatalo, H. Heiskanen, and P. Hämäläinen, “Predicting Game Difficulty and Churn Without Players,” Aug. 2020, doi: 10.1145/3410404.3414235.

[33] J. Li et al., A Difficulty-Aware Framework for Churn Prediction and Intervention in Games, vol. 1, no. 1. Association for Computing Machinery, 2021. doi: 10.1145/3447548.3467277.

[34] A. F. del Río, A. Guitart, and Á. Periánez, “A time series approach to player churn and conversion in videogames,” Intell. Data Anal., vol. 25, no. 1, pp. 177–203, 2021, doi: 10.3233/IDA-194940.

[35] B. Karmakar and P. Liu, “Improved retention analysis in freemium playing games by jointly modelling players ’ motivation, progression and churn,” no. July 2020, pp. 1–32, 2021, doi: 10.1111/rssa.12730.

[36] A. Guitart, P. Chen, and Á. Periáñez, “The Winning Solution to the IEEE CIG 2017 Game Data Mining Competition,” Mach. Learn. Knowl. Extr., vol. 1, no. 1, pp. 252–264, Dec. 2018, doi: 10.3390/make1010016.

[37] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-liehr, “Neurocomputing Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package),” Neurocomputing, vol. 307, pp. 72–77, 2018, doi: 10.1016/j.neucom.2018.03.067.