IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 1, 2022, accepted March 20, 2022, date of publication April 4, 2022, date of current version April 11, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164409

A Novel Method for IPTV Customer Behavior Analysis Using Time Series

TOMISLAV HLUPIĆ[1,2], DRAŽEN OREŠČANIN[1,2], AND MIRTA BARANOVIĆ[2], (Member, IEEE)

[1]Poslovna Inteligencija d. o. o., 10000 Zagreb, Croatia
[2]Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia

Corresponding author: Tomislav Hlupić (tomislav.hlupic@inteligencija.com)

This work was supported by the European Regional Development Fund under Grant KK.01.1.1.01.0009 (DATACROSS).

ABSTRACT Internet Protocol Television (IPTV) has had a significant impact on live TV content consumption in the past decade, as improvements in the broadband speed have allowed more data volume to be delivered. In addition to existing infrastructure, which is mostly based on the set top boxes, new content providers have emerged, utilizing newly developed proprietary streaming platforms. As the number of IPTV users grew, more volume and variety of data became available for analysis. By analyzing stored user actions, it is possible to create a multivariate time series that represents user behavior over time. The approach presented in the paper is based on multivariate time series generation from user data and determining the similarity between them. Time series are created for each user based on the proposed quantified action sets, grouped in the feature groups and summarized over time. The action sets and feature groups can be adjusted to a certain IPTV platform. The end result of the analysis is the similarity score matrix, generated by calculating the similarities of all users' time series, where the similarity measure calculation can be chosen arbitrarily.

INDEX TERMS IPTV, time series analysis, data analysis, user behavior analysis, time series similarity, user profiling.

I. INTRODUCTION

Time series of user-generated data are partially unpredictable for several reasons. One of them is user behavior, which might follow same patterns, but partly depends on various environmental impacts. Next to it, the circumstances in which the data are created are unknown, but still impact the behavior. Therefore, a conclusion can be drawn that the entire users' environment is dynamic. In the time series decomposition, the unrepeatable dynamic falls into the residual data which have a different impact on the time series analysis.

Time series analysis of the digital broadcasted content includes analyzing the customer related data (e. g. the stored actions of a certain user or a group of users), analysis of the channels on which the content was shown, analysis of the content level etc. The analysis' is done, among other usages, for content recommendation, personalization of the content for a certain customer, and churn predictions.

This paper presents a novel method for Internet Protocol Television (IPTV) user behavior analysis based on time series pattern detection. Time series are created from the discretized user actions (e.g., channel change, content search etc.) and their respective timestamps, forming an uninterrupted stream. The analysis focuses on detecting similarities in time series that can subsequently lead to the clustering of users with the same detected behavior.

The motivation behind this method is based on the observation of different IPTV users' behavior. While some users tend to focus on a certain content or type of content, others show a behavior called "channel zapping". Several approaches have been taken to identify user behavior, some of which are described in the Section II. Identification and quantification of user behavior is the foundation for user clustering, based on the calculated similarity of their behavior.

The user clusters later create the possibility of refining recommendations provided by the recommendation engine. Although some recommender systems have already been established in this domain, they mostly focus on comparing the users based on their watched content, rather than their behavior. Using clusters based on user behavior, the recommended content obtains a refined input of the content that users with similar behavior consume. This brings the possibility of, for instance, providing recommendations of different content types depending on users' activity

The associate editor coordinating the review of this manuscript and approving it for publication was Dost Muhammad Khan.

periods – which is significant for on-the-fly IPTV broadcasting. Another possibility of using the clusters is to deploy a different recommendation model for a single cluster. As the recommender systems rely on these data, their overview is included in related work.

While the proposed model exists as a standalone system, its usage is devised as a supplement for existing recommender systems. The model produces a similarity matrix of quantified users' behavior that acts as an input for the recommendation model, but also as a basis for additional analysis.

The main contributions of this paper can be summarized as follows:

- The digital broadcast content data are classified on a high level
- The approach of using the time series in the IPTV data analysis is considered and described
- A novel method for IPTV customer behavior analysis using time series is proposed and demonstrated

The remainder of this paper is organized as following: In Section II, an overview of prior related work in the field is presented. The Section III covers the topic of time series usage in the digitally broadcasted content analysis. It begins with an introduction to digital broadcast content data classification, and in the rest of the section, a novel method for IPTV customer behavior analysis using time series similarities is presented. In the section IV, the experimental work on the IPTV data set using proposed method is evaluated. Finally, in the Section V, the paper is concluded, along with a description of future work.

II. RELATED WORK

The research in the field of IPTV systems varies from introducing different recommender system models, by analyzing the user behavior up to analyzing the entire IPTV systems based on the user-generated data.

Recommender systems are well-known and widely used approaches for adjusting the content to the users' preferences. There are numerous models for generating recommendations based on previous content consumption data, such as collaborative filtering [1] [2], that mostly have to tackle the cold-start challenge [3]. Their implementation varies, depending on the distribution platform. A different approach has to be taken for digital terrestrial television models [4] compared to the video-on-demand models [5]. Several other approaches for IPTV recommendation model have been proposed, such as using transformed-based fusion [6] and time-context aware model based on Tensor learning [7]. Consumer feedback can also be explicit [5], which relies on the user providing explicit rating to the system.

Next to the collaborative filtering based recommender systems, content-based recommender systems [8] are also widely used. Content-based recommender systems typically use profile information filtering to create recommendations. In such systems, information from the consumer profile (age, gender, education, interests) and its content ratings are correlated with information about the content item itself and its various attributes. If the item has attributes similar to other items this consumer has rated highly, the system will recommend it to the user.

Another approach for building recommender systems are hybrid recommender systems, which combine both profile and interaction information [4], [9]. Hybrid recommender systems can have broader usage than recommending content, such as broadband data recommendations.

In order to support the recommender system in terms of adapting the system towards group-based recommendation, rather than the user-based ones, it is necessary to model the characteristics of the users, such as in [10]. The IPTV user behavior is a challenge addressed in research for over a decade, where the initial research [10], [11] was aimed at creating marketing strategies and detecting user activity peaks and channel zapping, which had an impact on the IPTV service performance. Moreover, an important aspect, especially for IPTV, is the channel popularity dynamics [13]. User feedback, mostly obtained as the implicit data based on user actions, provides valuable insight and the possibility of determining the user opinion of the broadcasted content [14], [15]. In [15], a framework for assessing the implicit feedback was proposed, based on tracking the change channel events [14], [16]. Based on this framework, the same group of authors built a model that utilizes implicit feedback and content metadata to classify viewers' opinions [17]. Another interesting approach, based on the hybrid trust metric was recently introduced in [18].

To some extent, [19] deals with streaming strategy classification which affects the user experience of content consumption. In the proposed framework, the authors classified the content into:

- time-shifted streaming (TSS), where users can access the content stored after it is created
- on-the-fly streaming (OFS), usually related to the retransmission of live programs

In addition, the broader classification given in [20] considers the type of services, which also includes the Video on Demand (VoD). From the perspective of the user behavior, the VoD platform follows different models with different inputs and is not a part of this research. Another study [21] focused on the IPTV platform, which had VoD and live TV content, and concluded that VoD holds the user activity longer, while live TV users tend to search for content by surfing through the channels.

Recently, a new approach to user behavior detection was addressed in [22] and [23]. In [22], the authors dealt with the channel zapping behavior of IPTV users, a feature that is present in the on-the-fly IPTV content broadcasting. The authors have split the user behavior into the

- watching session (period between turning on the TV, followed by active channel watching and turning off the TV)
- channel zapping type (three types of channel switching depending on the previous action)

- interesting channel watching (channel being watched longer than the threshold)
- transition between interesting channels (two or more sequential sessions of interesting channel watching)

Although this analysis focuses on the granularity of a channel, the zapping behavior is clearly detectable and useful for recommending purposes. The collected data were used to generate recommendations through six different recommendation systems, each of them focused on one particular type of information. Recommender scores are later combined using fusion functions. Finally, the authors presented the channel recommendation approach using an attention mechanism, which is used to improve the recommendation accuracy. This approach to observing an IPTV user behavior is similar to one proposed in this paper, although based on a finer granularity level of the user-related data.

Another approach to IPTV user behavior detection was described in [23], in which the authors have proposed a multi-item-sets fingerprinting to identify IPTV users. The proposed fingerprinting method is based on identifying both frequent individual activity items (FIA) and the frequent consecutive item sequences. As the user accounts can be shared by multiple actual users or even by unknown users, the authors dealt with the accuracy of the user identification. It was suggested that the statistical features of behavioral traces would be a more accurate approach. Moreover, a new algorithm was introduced to generate the feature vector, together with a new similarity distance.

All proposed approaches are embedded in a MISFUB [23] computing framework, which:

- uses the SURE algorithm [23] to construct the user digital behavior fingerprints from the sequence of items
- introduces the similarity distance using the Jaccard distance and a variant of the Kullback-Liebler divergence function
- introduces a fusion decision scheme to improve the performance of the algorithm and the similarity distance

The effectiveness of the new framework was demonstrated on a large IPTV subscriber dataset, with an average matching precision of 93.8% on the 1000 user dataset. Similar to the work presented in [22], the analyzed data granularity level was the selected channel.

An insight into the similarity measures at a high level is presented in [24] without any specific data context, presented for univariate time series. By using similarity measures, it is possible to classify the time series [25], which is a step toward creating time series clusters. In [26], a review of deep learning algorithms for time series classification was presented, which is a novel approach for using deep neural networks for time series classification. However, the multivariate time series present a significantly higher complexity when determining the similarity owing to their high dimensionality. Most approaches typically involve a variant of principal component analysis (PCA) [27]–[29]. One of recent approaches to clustering multivariate time series using common PCA was presented and evaluated in [28]. The aforementioned approach is an extension of a previous study [29] and can be evaluated for IPTV multivariate time series clustering. In [30], the authors focused their research on the correlation between simultaneous movement patterns of variables over time, with the multivariate pattern being a union of all univariate ordinal patterns. The recent work in [31] focused even more on the multivariate time series, as they emphasized human activity as a typical case with multiple observed dimensions. The main focus of this study was to evaluate the multivariate time series and state-of-the-art classifiers in a comprehensive overview.

In addition to similarity metrics, pattern recognition is another approach that can result in user behavior clustering [32], [33]. Pattern matching can be achieved by the moving-window approach, as described in [34], through template-based or rule-based approaches [35], or a combination of rule-based approaches and neural networks [36]. Time series motifs can also be used to create time series clusters using different approaches to similarity measures. A novel approach, which uses dynamic time warping (DTW) to measure the similarity between time series, was recently proven to have significant performance benefits over other methods, as presented in [37]. Dynamic time warping [38] is a well-known method that has a wide application on time series, such as finding patterns [39] and classification [40]. Pattern recognition in the scope of the IPTV user behavior analysis is planned as the next step in the research.

III. THE METHOD FOR IPTV CUSTOMER BEHAVIOR ANALYSIS USING TIME SERIES

A. CLASSIFICATION OF THE DIGITAL BROADCASTED CONTENT DATA

For analytical purposes, it is of utmost importance to classify the data according to their source, sort or usage. The IPTV data originate from various sources: they can be sourced on the provider's side, on the content creator's side or created by the service users.

The data created on the provider's side consist mostly of automatically generated technical data that act as a support for the broadcasted content. An example of technical data are extended metadata, which are bound to a certain content. Although usually smallest, volume-wise, their significance in analytics is rather important, as they are usually related to the data created on the user's side. Thus, a unique footprint of the link between user actions and the environment is generated, forming an insight into user behavior, thus forming the basis for recommender systems. Examples of these data are the current broadband speed, user package data (when they exist) etc.

Data created on the content generator side combine content-related metadata and the distribution-related metadata. Content-related metadata are rarely updated and are mostly focused on content, such as the name, content length, content type, genre etc., while content-specific data vary according to distribution type.

The highest volume data with the most variety are generated by the users, whose actions are stored as valuable data, providing insight not only on the users but also on the content. By combining the user action data with the content metadata, the data can be segmented over certain features or transformed into a time series. The time series enhance the static data, providing the possibility of additional time-variant data analysis. By creating time series from the features over time, it is possible to detect certain patterns in the user behavior or to predict how the users might react to content introduction in a given time slot.

TABLE 1. **Digital broadcasted content data features.**

Data source	Content provider	Content generator	User generated content
Feature			
Volume	Small to medium	Small	Large
Velocity	Medium to fast	Slow	Fast
Variety	Small to medium	Small	Large

Moreover, when considering digitally broadcasted content, it should be considered that content availability also has an impact on user behavior. The user's behavior is affected by the preference for the content availability, which has implications for further user classification. In addition, each of the content categories uses different prediction models in general, as the impact of live broadcasted content on user behavior and further recommendations has to have a different analytics approach than on-demand content.

Time series are defined as the sequence of discrete data in time and can be either univariate or multivariate. The multivariate time series is a sequence of pairs

$$X = [(p_1, t_1), \ldots, (p_i, t_i), \ldots, (p_n, t_n)] \times (t_1 < \ldots t_i < \ldots < t_n) \qquad (1)$$

where each p_i is a data point in a d-dimensional space, and each t_i is the time stamp at which p_i occurs [41].

In the context of classification, a time series is a list of vectors over d dimensions and m observations [31], denoted as

$$\boldsymbol{X} = \langle \boldsymbol{x}_1, \ldots, \boldsymbol{x} \rangle, \quad \text{where } \left(x_{1,k}, x_{2,k}, \ldots, x_{m,k}\right) \qquad (2)$$

We denote the jth observation of the ith case of dimension k as scalar $\mathcal{X}_{i,j,k}$.

Although time series analysis is mostly focused on future data forecasting, they can provide valuable insight into the data change over the time.

Time series used for digitally broadcasted content analysis can also be univariate or multivariate. Feature selection for the analysis is important for data granularity, as the data vary significantly in that segment. The most volume, velocity and variety-intense data with the greatest impact on the analysis are user-generated through certain user actions. These data must be tracked on the time granularity of a single second. The automatically generated data mostly lay in coarser granularity levels (sometimes on the daily level). Because these data are interdependent, first the data on the coarser granularity level must be multiplexed over the analyzed time frame in order to match the data on the common granularity.

An example of the analysis is the correlation between the user action, such as the content change or content search, and the recommendation shown during the live broadcast. If the recommended content is selected, the action is highly correlated with the previous recommendation. Another general example of the analysis is the time difference between the two content changes. The longer the difference, the higher the chance of the user's affinity towards a newly shown content.

The time series based on tracking the frequency of user actions (such as the channel change, defined in the next chapter) provides information on the users' behavior models. Users of similar behavior are later clustered, giving the content provider the possibility of generating various personally adjusted recommendations or recommendation groups.

As this work deals with the time series as a basis for user behavior detection, the other possible usages of the time series in IPTV will be mentioned as a part of future research. Forecasting, as the most common time series usage, will be particularly emphasized in it.

B. THE PROPOSED METHOD FOR IPTV CUSTOMER BEHAVIOR ANALYSIS

In this chapter, the proposed novel IPTV data analysis method is presented. The analysis method is based on developing the time series analysis, which serves as the framework for the similarity calculation. The algorithms for similarity, pattern matching and clustering are not a subject of this model and will be discussed in Section V.

The method is based on tracking user actions in a certain time frame, from which the multivariate time series representing the footprint of the user behavior is generated. The user actions, combined together, provide implicit feedback on the user's content interest that can be used for content recommendation purposes [2], [15]. All user actions are performed in the content consumption environment, which can be an STB, a dedicated application, a browser page, or any other available platform on which the content can be shown and consumed.

Definition 1: An action a is defined as a tuple consisting of the description and the related quantifier $a = (\delta, \theta)$, where the δ represents the action description, and θ represents the action quantifier in the time series. Action set A holds all the available actions $a_1, \ldots, a_n$.

An action is a time-independent event that can happen at an arbitrary point in time.

The actions typically come in pairs $\langle a_i, a_j \rangle$ which have opposite quantifier values, thus creating opposite results when performed. It should be noted that a single action can only belong to one action set.

A feature of the IPTV is a subset of the action set that holds the actions that move a certain IPTV platform component to a different state. Features are somehow dependent – some features can be prerequisites for others to occur (e.g. a channel change cannot be performed until the environment is turned on). However, each feature can be independently analyzed.

A data point p in time series X is a vector of a size d, where d denotes the number of observed dimensions [31]. A single feature is represented as a dimension of the analyzed time series.

Definition 2: An IPTV feature f is defined as a set of action pairs, $f = \left\{\langle a_{i1}, a_{j1}\rangle, \ldots, \langle a_{in}, a_{jn}\rangle\right\}$.

As the p_i is a vector, it can be represented as

$$p = \left[\langle f_1, q_1\rangle, \ldots, \langle f_d, q_d\rangle\right] \tag{3}$$

where each feature holds a certain quantifier value q.

Although it is mentioned in [30] that the features in the multivariate time series might be simultaneously dimension-dependent, it should be noted that in the proposed analysis method the features in p are collectively independent (with the exception of f_1, as the environment needs to be turned on before any other actions can be performed).

The actions, on which the method is based, are enumerated as shown in Table 2. It should be emphasized that the proposed action set is not the final action set and might be adapted depending on the IPTV platform or the available action set.

As the features are independent, their quantifiers can have different values. This is due to the impact each feature has on the analysis. For some features, it makes sense to decrease the value and efficiently set the time series value to the previous state. Features $f_1 \cdots f_4$ are the example of this representation, where the feature consists of actions that usually are of browsing type (except f_1 in the proposed feature set). Feature f_1 represents the state of the user's content consumption environment (STB, application etc.).

On the other hand, the feature f_5 represents a different action that, through direct search (channel selection through the channel number, content search through the name or other keyword etc.) sets the environment to the search state followed by either another search, content consumption or environment shutting down. Therefore, the consumption environment is either in the search state or in the consumption state, and feature f_5 represents those states.

An IPTV platform is a content delivery system that provides digitally broadcasted content over the Internet protocol to users. In theory, the platform can provide the access to a finite number of users n, which is limited by the hardware, software and broadband constraints. Each user u_i represents an independent subscriber to the IPTV platform who accesses the content through a hardware client (STB) or a dedicated application. Through user interaction with either a client or an application, a log of the interaction data is generated. Interaction data can contain, for example, user action, previously consumed channel, or a chosen consumed content.

TABLE 2. Action sets and Quantifier Values.

Action description	Quantifier	Feature description	Feature mark
STB ON	1	Service state	f_1
STB OFF	-1	Service state	f_1
Service ON	1	Service state	f_1
Service OFF	-1	Service state	f_1
Channel FORWARD	1	Content browse	f_2
Channel BACKWARD	-1	Content browse	f_2
Content item FORWARD	1	Content browse	f_2
Content item BACKWARD	-1	Content browse	f_2
EPG channel FORWARD	1	Content description browse	f_3
EPG channel BACKWARD	-1	Content description browse	f_3
EPG channel timeline FORWARD	1	Content description browse	f_3
EPG channel timeline BACKWARD	-1	Content description browse	f_3
EPG info ON	1	Content list state	f_4
EPG info OFF	-1	Content list state	f_4
Content list ON	1	Content list state	f_4
Content list OFF	-1	Content list state	f_4
Search START	1	Search group	f_5
Search END	0	Search group	f_5

By joining the action result (e.g., chosen channel/content) to the action, a valuable data on the user affinity are created. Therefore, a single action a_i, with the accompanying result vector r_i and quantified state vector ω_i, exists for user u_i at a certain moment of time t. The quantified state vector ω_i holds the sum of all quantifiers for each of the actions that have occurred by the moment t. Together, they represent a state of user u_i on the IPTV platform in a moment t, defined as

$$\mathrm{P}_i(t) = (a_i, r_i, \omega_i(t)), \quad i = 1, \ldots, n \quad \text{where } n \text{ denotes the number of users of the IPTV platform} \tag{4}$$

The beginning of the time series t_s is set arbitrarily; it can either the active user's time using the platform or can use the longer time frame where the platform inactive time is quantified as 0. When they are used, the service state set feature explicitly sets the active timeframe boundaries based on the defined user actions. The end of the time series t_e, either defined explicitly or implicitly, effectively ends the observed period.

Even though the time series are theoretically unbounded, in order to achieve the possibility of comparing different users' behavior, the starting timestamp and the ending timestamp of the compared period must be aligned. Usually, the behavior is tracked on a daily level, so the typical boundaries would be $t_s = 00:00:00$ and $t_e = 23:59:59$. However, for the prediction applications of the time series, the boundaries $[t_s, t_e]$ may stretch over a longer time period.

Therefore, the dataset from which the time series is created can be represented as:

$$\mathcal{P} = \{\mathrm{P}_i(t), i = 1, \ldots, n; t = t_s, \ldots, t_e\} \tag{5}$$

In each time stamp t of the time series, each P_i has to have the current quantified state vector $\omega_i(t)$ stored. The $\omega_i(t)$ is calculated as the sum of all the previous action quantifiers that occurred by t for a certain action a_i and can be denoted as

$$\omega_i(t) = [(a_1, \varphi_1), \ldots, (a_n, \varphi_n)], \quad a_1 \cdots a_n \in A \tag{6}$$

where φ represents the quantified state for each action a. For action a, at a moment of time t, the quantified state φ holds the sum of all quantifiers of occurrences of the given action.

Algorithm 1 Calculating the quantified state vector $\omega_i(t)$

```
Input: $a_i, A, t, \omega_i(t-1)$
Output: $\omega_i(t)$
Function $\omega_i(t) = \text{value}(a, A, t, \omega_i(t-1))$
For each $a \in$ A
    If $a = a_i$
        Get index $j$ of $a_i$ in $\omega(t)$
        Get $\varphi_j$ for $a_j$ from $\omega(t-1)$
        $\varphi_j = \varphi_j + \theta_j$
    End if
End for
```

In a single moment of time t, as the algorithm points out, only one update of $\omega_i(t)$ will occur.

The algorithm can be summed as following:

For each user, the quantified state vector ω_i changes over time. A single action per user might occur in a single moment in time. If action a_i is detected, the algorithm searches for the action's index j inside vector ω_i, in order to update the state φ_j related to the detected action. Only a single value is updated at each moment, by adding the quantifier value to the previous state for the detected action. The other values in the vector are skipped.

A multivariate time series can be seen as a multiple univariate time series in a d-dimensional space, in which a single dimension corresponds to a certain feature f. The feature data are generated from aggregated quantified state data of all the action pairs $\{\langle a_{i1}, a_{j1}\rangle, \ldots, \langle a_{in}, a_{jn}\rangle\}$, to which they belong.

A multivariate time series X_i denotes the time series representing the discretized behavior of user u_i.

Algorithm 2 Building the multivariate time series X_i from $\mathcal{P}$

```
Input: $\mathcal{P}, A, t_s, t_e, i$
Output: $\mathrm{X}_i$
Function $\mathrm{X}_i = \mathrm{MVTS}(\mathcal{P}, A, t_s, t_e, i)$
For each $t$ between $t_s$ and $t_e$
    For each $f$ in $p$
        Set $q = 0$
        If $a \in f$
            Then extract $\varphi$ for $a$ from $\omega(t)$
            $q = q + \varphi$
        End If
    End for
End for
```

The described algorithm generates a multivariate time series X_i consisting of d dimensions, where each dimension represents a single tracked feature f. The time series is built with the boundaries of t_s and t_e. In each timestamp t, the state $\mathrm{P}_i(t)$ is observed. For each feature, the state of adjoined action pairs is summed to determine the value of the feature in an observed timestamp t.

The set of all the time series for the n users is denoted as

$$TS = \{\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n\} \tag{7}$$

Using the common intervals proposed in [24], the similarity score s between two time series is represented using a value in the interval [0, 1], where 1 is the value that represents the maximum similarity of two time series. Each X_i is compared to the members of the set $TS \setminus \mathrm{X}_i$, resulting in the $n-1$ pairs of values s^i_j where j represents the time series of user u_j.

The final result of the analysis is the matrix of the similarities between users, generally represented as a matrix of similarity scores.

$$S = \begin{pmatrix} 1 & s^1_2 & \ldots & s^1_j & \ldots & s^1_n \\ s^2_1 & 1 & \ldots & s^2_j & \ldots & s^2_n \\ \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ s^i_1 & s^i_2 & \ldots & 1 & \ldots & s^i_n \\ \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ s^m_1 & s^m_2 & \ldots & s^m_j & \ldots & 1 \end{pmatrix} \tag{8}$$

IV. EXPERIMENT

In this chapter, the results of applying the algorithm to the test dataset are presented and compared with similar previous studies. The experiment is divided into two separate tasks. The first is creating time series from the dataset for a single user and a single feature. The second is building the similarity matrices for the subset of users, as a basis for further user clustering based on their behavior footprint. Finally, the proposed method is compared with different IPTV user behavior analysis approaches.

The data provided are the testing data previously used for building IPTV recommender systems for a small number of users. All users are represented with their identifier in the system, which is conveniently converted into appropriate tags to further anonymize the data. The actions provided by the data are limited, so only three features (f_2, f_3 and f_4) from the proposed feature set could have been applied, as the service state and search group data were omitted.

A. TIME SERIES CREATION

Initially, the data are represented with tuples consisting of four values: the user identifier, the channel identifier that resulted from the action, the applied action identifier and the timestamp. Out of these values, the channel data does not have an impact on the algorithm, but holds valuable data that can be used for additional analysis. Prior to the algorithm application, the data had to be cleansed and prepared by adjusting the data for time series analysis. This was done by indexing the records by their timestamp and calculating the quantified state vector $\omega_i(t)$ in the given moment of time t, as proposed in Algorithm 1.

The final result is a set of four time series for each user: a time series, built as proposed in the Algorithm 2, represents the multivariate time series X_i. Of these, three separate time series, one representing each feature, are extracted as they can provide focused insight for further analysis. The visual representations of the time series for the three different users are shown in the Figures 1, 2 and 3.

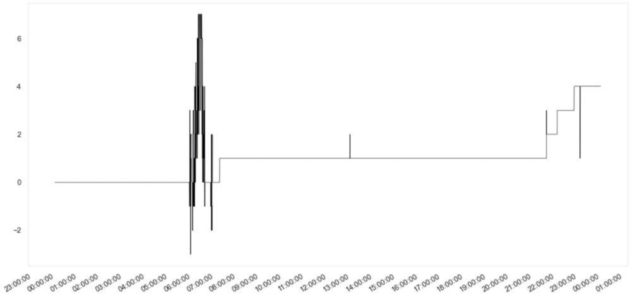


FIGURE 1. User u_5 with low activity.

These three users show a completely separate behavior footprint: in the first figure (Figure 1), representing the activity of the user u_5, it is visible that the user has some activity in the morning with quick channel browsing inside a single hour and some focused activity in the evening hours. Apart from that, there is no activity throughout the day.

The second figure (Figure 2), representing the user u_1, shows the user with a high rate of channel zapping during the entire day. This behavior indicates a low focus on the content and high activity engagement, so this user can be a candidate for a separate recommender system that does not take content into consideration.

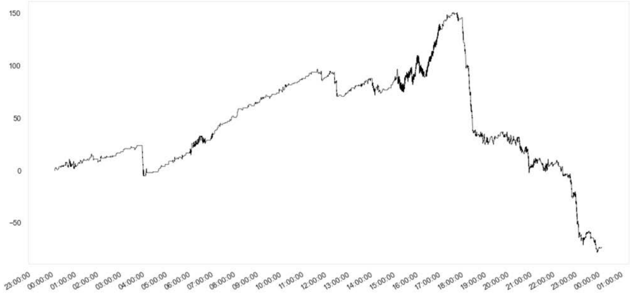


FIGURE 2. User u_1 with high activity.

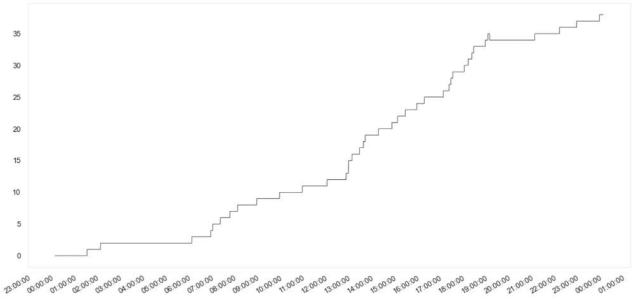


FIGURE 3. User u_6 with focused activity.

In the third figure (Figure 3), a representation of moderate activity during most of the day, with a focus on specific content, is shown. This figure represents user u_6, which is a good candidate for recommender models built around content recommendations.

B. SIMILATIRY MATRIX CREATION AND ANALYSIS

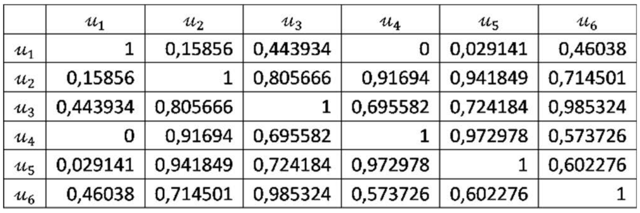
The second part of the experiment involve calculating the user behavior similarity using dynamic time warping through its implementation FastDTW [42]. As mentioned in the related work, dynamic time warping is widely used in the time series calculations. It is highly applicable in the case of comparing the time series built with the proposed algorithm as it takes time shifting into consideration. This is valuable as users might have similar behavior in the different time slots, so the dynamic time warping detects their similarity better than algorithms such as Euclidian distance.

For the experiment purposes, different time series are created for a random user sample. The similarity of the time series is represented as a matrix of $n \times n$ size, where n is the number of compared users. In this experiment, six users are compared, so the matrices are 6×6 in size. The interdependence of the values in the matrix is explained in the previous section. Each matrix is the result of comparing the time series of the same length using dynamic time warping.

After the calculations and initial matrix creation, the values in a single matrix are normalized using min/max normalization. The main reason for normalization is to represent the similarity as a value that is more suitable for analysis. The final similarity values fall in the range between 0 and 1, where 1 represents identical user behavior and 0 represents the most diverse user behavior in the matrix.

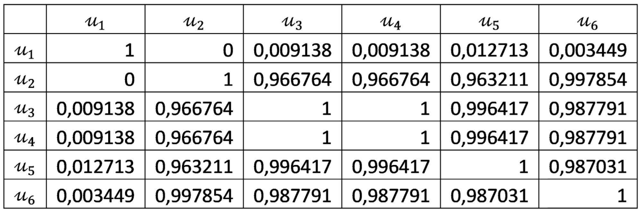
The first group of time series represents the user behavior over a time span of eight days. In this group, a user behavior is represented by three time series, each representing a certain feature (f_2, f_3, f_4). For each feature, a separate, independent similarity matrix is created.

As is visible from the first matrix, user u_1 has a significantly more divergent behavior than the rest of the analyzed users. By taking into consideration a sample of the user behavior shown in Figure 1, the result of the matrix related to the feature f_2 is expected, as the user has a significantly higher volume of actions and channel zappings than other users. Even more disparate behavior is shown for feature f_3, as the user was using EPG browsing more than other analyzed users. The last feature, f_4, representing the EPG service start and stop actions, shows less, but still significant disparity of behavior.



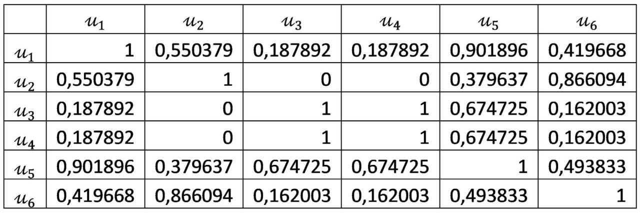
	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0,15856	0,443934	0	0,029141	0,46038
u_2	0,15856	1	0,805666	0,91694	0,941849	0,714501
u_3	0,443934	0,805666	1	0,695582	0,724184	0,985324
u_4	0	0,91694	0,695582	1	0,972978	0,573726
u_5	0,029141	0,941849	0,724184	0,972978	1	0,602276
u_6	0,46038	0,714501	0,985324	0,573726	0,602276	1

FIGURE 4. Similarity matrix for f_2 for eight-day time span.



	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0	0,009138	0,009138	0,012713	0,003449
u_2	0	1	0,966764	0,966764	0,963211	0,997854
u_3	0,009138	0,966764	1	1	0,996417	0,987791
u_4	0,009138	0,966764	1	1	0,996417	0,987791
u_5	0,012713	0,963211	0,996417	0,996417	1	0,987031
u_6	0,003449	0,997854	0,987791	0,987791	0,987031	1

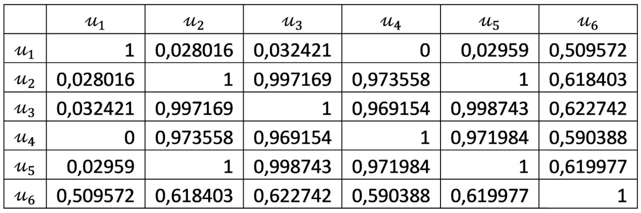
FIGURE 5. Similarity matrix for f_3 for eight-day time span.



	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0,550379	0,187892	0,187892	0,901896	0,419668
u_2	0,550379	1	0	0	0,379637	0,866094
u_3	0,187892	0	1	1	0,674725	0,162003
u_4	0,187892	0	1	1	0,674725	0,162003
u_5	0,901896	0,379637	0,674725	0,674725	1	0,493833
u_6	0,419668	0,866094	0,162003	0,162003	0,493833	1

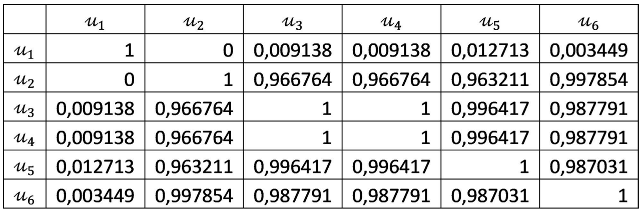
FIGURE 6. Similarity matrix for f_4 for eight-day time span.

The second group of time series consists of user actions during one weekend. As in the previous group, each feature for a single user is represented as an independent similarity matrix.



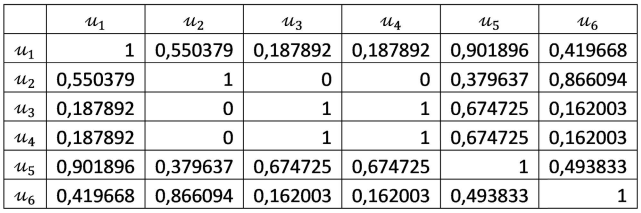
	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0,028016	0,032421	0	0,02959	0,509572
u_2	0,028016	1	0,997169	0,973558	1	0,618403
u_3	0,032421	0,997169	1	0,969154	0,998743	0,622742
u_4	0	0,973558	0,969154	1	0,971984	0,590388
u_5	0,02959	1	0,998743	0,971984	1	0,619977
u_6	0,509572	0,618403	0,622742	0,590388	0,619977	1

FIGURE 7. Similarity matrix for f_2 for a single weekend time span.



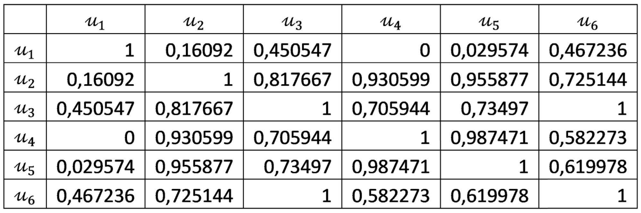
	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0	0,009138	0,009138	0,012713	0,003449
u_2	0	1	0,966764	0,966764	0,963211	0,997854
u_3	0,009138	0,966764	1	1	0,996417	0,987791
u_4	0,009138	0,966764	1	1	0,996417	0,987791
u_5	0,012713	0,963211	0,996417	0,996417	1	0,987031
u_6	0,003449	0,997854	0,987791	0,987791	0,987031	1

FIGURE 8. Similarity matrix for f_3 for a single weekend time span.



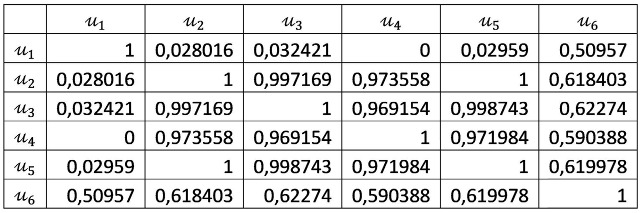
	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0,550379	0,187892	0,187892	0,901896	0,419668
u_2	0,550379	1	0	0	0,379637	0,866094
u_3	0,187892	0	1	1	0,674725	0,162003
u_4	0,187892	0	1	1	0,674725	0,162003
u_5	0,901896	0,379637	0,674725	0,674725	1	0,493833
u_6	0,419668	0,866094	0,162003	0,162003	0,493833	1

FIGURE 9. Similarity matrix for f_4 for a single weekend time span.



	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0,16092	0,450547	0	0,029574	0,467236
u_2	0,16092	1	0,817667	0,930599	0,955877	0,725144
u_3	0,450547	0,817667	1	0,705944	0,73497	1
u_4	0	0,930599	0,705944	1	0,987471	0,582273
u_5	0,029574	0,955877	0,73497	0,987471	1	0,619978
u_6	0,467236	0,725144	1	0,582273	0,619978	1

FIGURE 10. Similarity matrix for all features for eight-day time span.



	u_1	u_2	u_3	u_4	u_5	u_6
u_1	1	0,028016	0,032421	0	0,02959	0,50957
u_2	0,028016	1	0,997169	0,973558	1	0,618403
u_3	0,032421	0,997169	1	0,969154	0,998743	0,62274
u_4	0	0,973558	0,969154	1	0,971984	0,590388
u_5	0,02959	1	0,998743	0,971984	1	0,619978
u_6	0,50957	0,618403	0,62274	0,590388	0,619978	1

FIGURE 11. Similarity matrix for all features for a single weekend time span.

Compared to the results of the eight-day time span analysis for the f_2 (Figure 4), the diversity between user u_1 and other users is even greater. Simultaneously, the behaviors of users u_2, u_3 and u_5 are almost identical, whereas user u_4 shows very close similarity. In addition, the difference in weekend behavior of users u_3 and u_6 is significantly greater than that during the eight-day time span (Figure 7).

By this comparison, it is clear that some users exhibit different behaviors during the weekends. This, for instance, provides the possibility of treating the recommendations differently during weekdays and weekends for these users.

Another valuable insight is the comparison of matrices for features f_3 and f_4. By having identical values in these matrices, it can be concluded that users show almost no difference in the behavior during the longer time span (Figures 5 and 6) and weekends (Figures 8 and 9). In this case, it would be opportune to omit the time series of the user with the greatest behavior difference from the analysis so that the other users can be analyzed closely.

The third group of similarity matrices has a different basis than the previous two. For this group, the dynamic time warping algorithm is applied to a multivariate time series with all three features belonging to a single time series. Two time series for each user are created: one holding the data of the eight-day time span (Figure 10), and another holding the data of a single weekend (Figure 11).

In the case of this user group, the similarity of these two matrices with the ones representing the feature f_2 in the same timespan shows that this feature has the most impact in the multivariate analysis. Even so, having a separate analysis of other features can be beneficial as they identify other behavior characteristics – for example - users that use EPG more tend to be more content oriented.

C. IPTV TIME SERIES CLUSTERING

The next step in the analysis is the clustering of the created time series, in order to detect users with similar behavior on the larger scale. The dataset on which the clustering is performed contains a week-long data of the users' actions, transformed into time series using the proposed algorithm.

The clustering is performed using self-organizing maps (SOM) [43], a neural network that utilizes unsupervised learning process to produce classes of patterns. With addition to existing users and their respective time series, twelve more users are added to the analysis, with their time series built for f_2, as shown on the Figure 12. The main reason for the usage of only one feature is to reduce the dimensionality to the most significant feature, that was previously proven to be f_2.

Although k-means is the usual choice for unsupervised learning, SOM was proven in [44] to produce the same results while outperforming k-means and having less variations in results.

During preprocessing, the calculated time series are normalized using min-max normalization. The normalized values are used as an input for the SOM, with the number of clusters being manually set to four. It should be emphasized that the initial number of clusters is set arbitrarily as a part of the research, and the number itself depends on the nature of the analysis.

As a result of first clustering and clusters visualization, shown on the Figure 13, it can be concluded that the users with similar weekly behavior are clustered together. Cluster 1 shows that the users with moderate activity during the working days and higher activity starting on Friday are separated from the users with only weekend activity (Cluster 3) and users that are active in the middle of the week (Cluster 4).

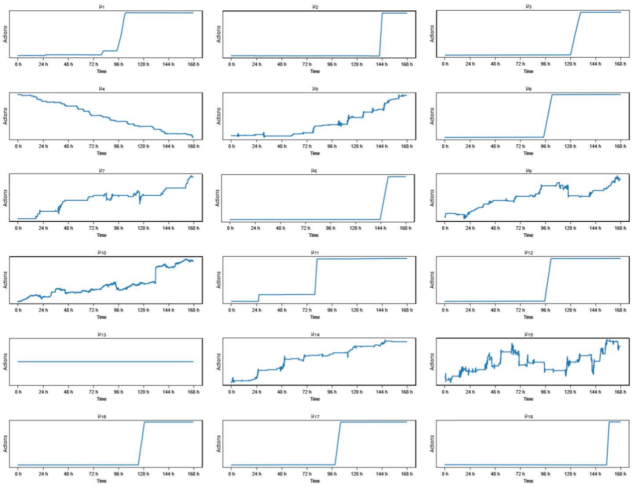


FIGURE 12. The analyzed users and their respective time series.

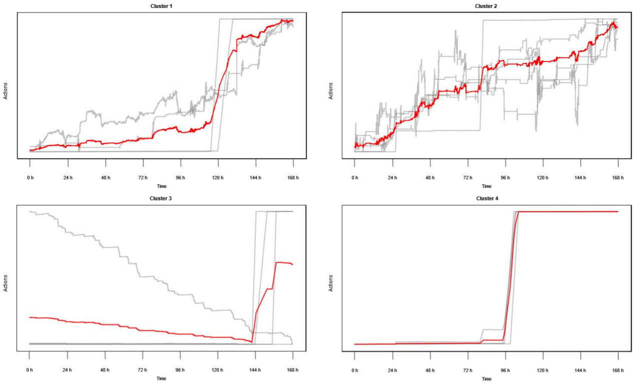


FIGURE 13. Four determined clusters with their respective averages.

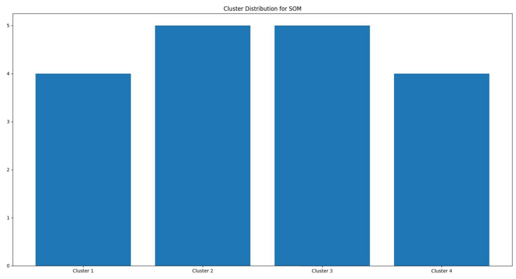
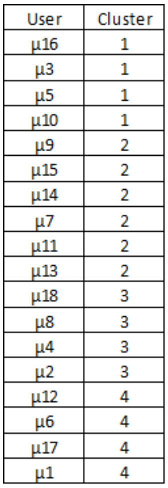


FIGURE 14. Cluster distribution for SOM with four clusters.

The highly active users, with one outlier, are grouped in the Cluster 2. The distribution of the users' time series is shown on the Figures 14 and 15.

Typically, users that have a behavior pattern similar to the one in the Cluster 4 are highly focused on a small number of content activities that are the main reason for the content consumption in the first place. In this cluster, it would be recommendable to analyze the consumed content, e.g. sport events, and tailor the recommendations accordingly. Unlike this cluster, for the clusters 1 and 2 the similar approach



User	Cluster
µ16	1
µ3	1
µ5	1
µ10	1
µ9	2
µ15	2
µ14	2
µ7	2
µ11	2
µ13	2
µ18	3
µ8	3
µ4	3
µ2	3
µ12	4
µ6	4
µ17	4
µ1	4

FIGURE 15. **Distribution of users' time series in four clusters.**

cannot be applied, as these users' activity is less driven by the content, and more by the consumption regardless of the content.

Through observing the initial clustering, it is determined that several time series should fit better in their own cluster due to calculated similarity being closer to a certain cluster. By raising a number of clusters to six, the time series show more precise cluster fit.

The result of changing the number of clusters to six is shown on Figures 16, 17 and 18.

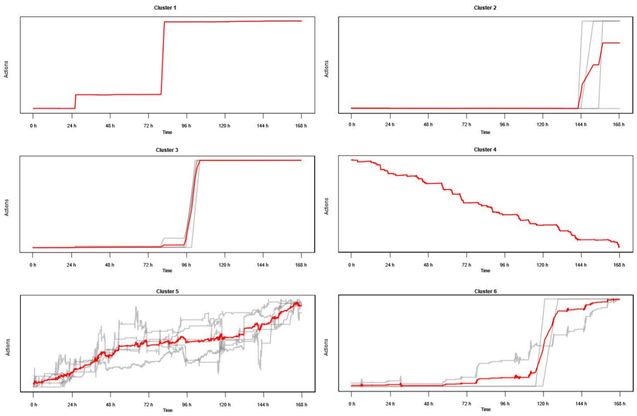


FIGURE 16. **Six determined clusters with their respective averages.**

The difference is especially seen on clusters 2, 3 and 6 where the less active users are clustered together based on the start of the higher weekly activity. Moreover, a user that has a pattern of channel change through using channel backward action is detached into its own cluster. The highly active users with apparent channel zapping behavior are clustered together in the cluster 5.

By having users clustered together as in the output of this experiment, an opportunity exists for application of the recommendation algorithm on a smaller number of users with a similar behavior. This can lead both to recommendations that are more precise and execution performance gains, as fewer comparisons are needed.

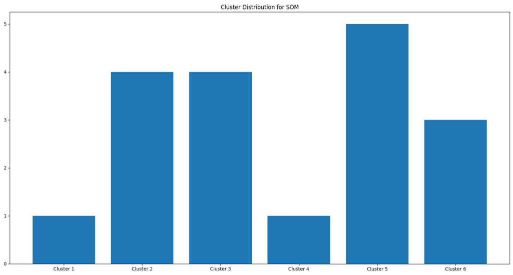
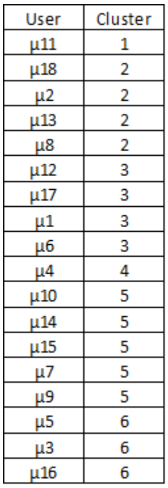


FIGURE 17. **Cluster distribution for SOM with six clusters.**



User	Cluster
µ11	1
µ18	2
µ2	2
µ13	2
µ8	2
µ12	3
µ17	3
µ1	3
µ6	3
µ4	4
µ10	5
µ14	5
µ15	5
µ7	5
µ9	5
µ5	6
µ3	6
µ16	6

FIGURE 18. **Distribution of users' time series in four clusters.**

D. COMPARISON WITH OTHER APPROACHES

The proposed method differs significantly from the other approaches that are oriented towards IPTV user behavior. The most similar approach recently was done by [17], where the authors focused on the implicit user feedback through their actions (channel zapping etc.) but also on the explicit actions – that the proposed method did not take into consideration. The authors' approach was focused on building a model between explicit and implicit ratings and using consumed content as another dimension in the analysis.

The method proposed in this paper omits the explicit feedback and content as a dimension that is consumed in [17] and focuses only on the detected behavior. This is done to provide the basis for further user clustering based on their calculated behavior and the approach presented in [17] could be applied once the users are already clustered.

Another approach that was focused on a holistic analysis of the IPTV user behavior was presented in the paper [12], where the authors focused on detecting certain action patterns in the system, such as channel zappings and uninterrupted content consuming sessions. This approach proved to be more oriented towards the analysis of the entire system and all users together, rather than on the individual user. Several other papers also focus mostly on a certain pattern detection in the system, without including other IPTV-related actions (content time shifting and browsing, etc.) and quantifying them towards the user behavior description.

E. APPLICATIONS IN IPTV SERVICES AND RECOMMENDER SYSTEMS

The proposed method is mostly aimed at live TV systems, where the users have the possibility of faster content switching. The video-on-demand IPTV systems typically have less user interaction with more focus on the content, so the time series would merely show the activity periods. In the live TV systems, the users show much more divergent behavior patterns, as more user actions are available.

By introducing user behavior quantification and its representation through time series, the users can be clustered based on their behavior. The IPTV providers can benefit from known user clusters as it provides an opportunity for narrower detection of the cluster's content and activity affiliation. An example would be the indication of clusters with typical activity on Tuesday to Thursday evening with little or no activity during other days. These clusters can then be analyzed from the content perspective and can indicate the users that consume sports content, such as continental club football competitions. Moreover, this would omit the users with potentially less interest in the same content from the analysis, thus creating smaller data subsets that are analyzed. By detecting information like this, the IPTV providers can adjust the subscription packages to match the detected users' affinities.

Further on, the impact on the recommender system comes from the time series clustering. The currency recommender systems rely on algorithms such as collaborative filtering that require the processing of all the data in the system to detect similarities in the consumed content. This approach does not consider user behavior but relies solely on content consumption. Through introducing user behavior quantification, represented through the time series, the clusters of users with similar quantified behavior are detected. Therefore, the recommendation algorithm ran over a smaller cluster of users already has some similarities detected, the detection of similar content is done faster and potentially more accurately.

Recommendations in live IPTV have to be time-sensitive, which is a dimension usually omitted in most of the recommendation systems. The time series, created from the user actions carrying the timestamp, provides the time data that can narrow down the user activity periods. Having the defined periods in which the user consumes the content, the recommendation systems can recommend the live content that falls in those time slots.

V. CONCLUSION AND FUTURE RESEARCH

In this paper, a method for the IPTV customer behavior determination and analysis is presented, based on the generation of multivariate time series. The method is based on quantifiable actions that are grouped in pairs and tracked as a feature of the IPTV platform. Each multivariate time series represents a d-dimensional footprint of user behavior within a certain time frame. In each timestamp, the IPTV platform is set to a certain state, presented through a dedicated algorithm, for each user. Based on the state values for a certain trackable action, a dimension metric that represents an IPTV platform feature is calculated using the proposed time series generation algorithm. Finally, a similarity matrix is generated by comparing the generated time series for all IPTV platform users.

The method presented in this paper is the basis for generating user clusters with similar IPTV platform usage behavior. By clustering these users, it is possible not only to determine their behavior similarity, but also to describe it further through manual or automatic analysis. This can affect the recommendation system in a certain way, as the recommendations for the different user clusters might have distinctive, more suitable models that can be further refined and adjusted.

An approach comparable to the use of similarity measures can be applied to detect patterns in user behavior. In this case, the generated time series would not be compared to another user's time series; the pattern matching algorithms replace the similarity calculation.

The goal of introducing an analysis approach based on the actions is not to supplant the existing content-based recommendation models. The content-based recommendations and actions-based analysis should complement each other, resulting in content-based and behavior-based recommendations.

In addition to user clustering, the other possibility of using the generated time series is user behavior forecasting. As time series are typically used in forecasting, the recommendation algorithms can be validated by checking whether the recommended content falls into the timeline of the detected user behavior. Therefore, the recommended content can be further refined.

Future research will primarily focus on algorithm result storage and graph analysis of the results. In addition, the similarity metrics calculation will be tested with special consideration of analysis dimensionality using the state-of-the-art approaches. Next to the similarity metrics, various approaches of pattern matching will be evaluated, in order to create the basis for user clustering.

ACKNOWLEDGMENT

The authors would like to thank the Poslovna inteligencija, especially Goran Gvozden, Ph.D. and Marko Štajcer, who continuously gave their valuable advice, and also would like to thank the anonymous reviewers for their suggestions that led to improvements of the paper.

DISCLOSURE

No potential conflict of interest was reported by the author(s).

REFERENCES

[1] D. Orescanin, T. Hlupic, and I. Soric, "Predictive models for digital broadcasting recommendation engine," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectroon. (MIPRO)*, May 2018, pp. 1243–1248.

[2] S. Sidana, M. Trofimov, O. Horodnytskyi, C. Laclau, Y. Maximov, and M.-R. Amini, "User preference and embedding learning with implicit feedback for recommender systems," *Data Mining Knowl. Discovery*, vol. 35, no. 2, pp. 568–592, Mar. 2021.

[3] P. Cremonesi and R. Turrin, "Analysis of cold-start recommendations in IPTV systems," in *Proc. 3rd ACM Conf. Recommender Syst. (RecSys)*, 2009, pp. 233–236.

[4] H. P. Tukuljac, D. Nad, G. Stupar, and M. Z. Bjelica, "A solution of a DTV recommendation engine based on broadband and broadcast data," in *Proc. 22nd Telecommun. Forum Telfor (TELFOR)*, Nov. 2014, pp. 893–896.

[5] Y.-D. Seo, E. Lee, and Y.-G. Kim, "Video on demand recommender system for internet protocol television service based on explicit information fusion," *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113045.

[6] H. Li, H. Lei, M. Yang, J. Zeng, D. Zhu, and S. Fu, "A transformer-based fusion recommendation model for IPTV applications," in *Proc. 3rd Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2020, pp. 177–182.

[7] X. Yin, Y. Chen, X. Mi, H. Wang, Z. Tang, C. He, and D. Fang, "Time context-aware IPTV program recommendation based on tensor learning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[8] S. Song, H. Moustafa, and H. Afifi, "Advanced IPTV services personalization through context-aware content recommendation," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1528–1537, Dec. 2012.

[9] E. Amolochitis, I. T. Christou, and Z.-H. Tan, "Implementing a commercial-strength parallel hybrid movie recommendation engine," *IEEE Intell. Syst.*, vol. 29, no. 2, pp. 92–96, Mar. 2014.

[10] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao, "Modeling user activities in a large IPTV system," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf. (IMC)*, 2009, pp. 430–441.

[11] B. Xiao, J. Yan, X. Guo, and L. Leung, "IPTV: User behavior analysis," in *Proc. Int. Conf. Manage. Service Sci.*, Sep. 2009, pp. 1–4.

[12] G. Yu, T. Westholm, M. Kihl, I. Sedano, A. Aurelius, C. Lagerstedt, and P. Odling, "Analysis and characterization of IPTV user behavior," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, May 2009.

[13] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, and J. Xu, "Modeling channel popularity dynamics in a large IPTV system," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 1, pp. 275–286, Jun. 2009.

[14] M. Kren, U. Sedlar, J. Bester, and A. Kos, "Determination of user opinion based on IPTV data," in *Proc. 18th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2016, pp. 1–5.

[15] M. Kren, A. Kos, Y. Zhang, A. Kos, and U. Sedlar, "Public interest analysis based on implicit feedback of IPTV users," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2077–2086, Aug. 2017.

[16] M. Kren, A. Kos, and U. Sedlar, "Mining the IPTV channel change event stream to discover insight and detect ads," *Math. Problems Eng.*, vol. 2016, p. 5, Mar. 2016, Art. no. 2541814.

[17] M. Kren, A. Kos, and U. Sedlar, "Modeling opinion of IPTV viewers based on implicit feedback and content metadata," *IEEE Access*, vol. 7, pp. 14455–14462, 2019.

[18] H. Wang, D. Chen, and J. Zhang, "Group recommendation based on hybrid trust metric," *Automatika*, vol. 61, no. 4, pp. 694–703, Oct. 2020.

[19] M. Masciopinto, P. Comesaña, and and F. Pérez-González, "IPTV streaming classification," in *IPTV Delivery Networks*. Hoboken, NJ, USA: Wiley, 2018, pp. 25–63.

[20] A. Punchihewa and A. M. De Silva, "Tutorial on IPTV and its latest developments," in *Proc. 5th Int. Conf. Inf. Autom. Sustainability*, Dec. 2010, pp. 45–50.

[21] N. Liu, H. Cui, S.-H.-G. Chan, Z. Chen, and Y. Zhuang, "Dissecting user behaviors for a simultaneous live and VoD IPTV system," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 10, no. 3, pp. 1–16, Apr. 2014.

[22] G. Li, L. Qiu, C. Yu, H. Cao, Y. Liu, and C. Yang, "IPTV channel zapping recommendation with attention mechanism," *IEEE Trans. Multimedia*, vol. 23, pp. 538–549, 2021.

[23] C. Yang, L. Wang, H. Cao, Q. Yuan, and Y. Liu, "User behavior fingerprinting with multi-item-sets and its application in IPTV viewer identification," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2667–2682, 2021.

[24] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata, and A. Pulvirenti, "Similarity measures and dimensionality reduction techniques for time series data mining," in *Advances in Data Mining Knowledge Discovery and Applications*. London, U.K.: IntechOpen, 2012.

[25] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance based time series classification," in *Data Mining Knowl. Discovery*, vol. 33, no. 2, pp. 378–412, 2019.

[26] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Mar. 2019.

[27] K. Yang and C. Shahabi, "A PCA-based similarity measure for multivariate time series," in *Proc. 2nd ACM Int. Workshop Multimedia Databases (MMDB)*, 2004, pp. 65–74.

[28] H. Li, "Multivariate time series clustering based on common principal component analysis," *Neurocomputing*, vol. 349, pp. 239–247, Jul. 2019.

[29] H. Li, "Accurate and efficient classification based on common principal components analysis for multivariate time series," *Neurocomputing*, vol. 171, pp. 744–753, Jan. 2016.

[30] M. Mohr, F. Wilhelm, M. Hartwig, R. Möller, and K. Keller, "New approaches in ordinal pattern representations for multivariate time series," in *Proc. 33rd Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, 2020, pp. 124–129.

[31] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: A review and experimental evaluation of recent algorithmic advances," *Data Mining Knowl. Discovery*, vol. 35, no. 2, pp. 401–449, Mar. 2021.

[32] S. Spiegel, J. Gaebler, A. Lommatzsch, E. De Luca, and S. Albayrak, "Pattern recognition and classification for multivariate time series," in *Proc. 5th Int. Workshop Knowl. Discovery Sensor Data (SensorKDD)*, 2011, pp. 34–42.

[33] J. Lin, S. Williamson, K. Borne, and D. DeBarr, "Pattern recognition in time series," in *Advances in Machine Learning and Data Mining for Astronomy*. Boca Raton, FL, USA: CRC Press, 2012, pp. 617–645.

[34] A. Singhal and D. E. Seborg, "Pattern matching in multivariate time series databases using a moving-window approach," *Ind. Eng. Chem. Res.*, vol. 41, no. 16, pp. 3822–3838, 2002.

[35] T.-C. Fu, F.-L. Chung, R. Luk, and C.-M. Ng, "Stock time series pattern matching: Template-based vs. rule-based approaches," *Eng. Appl. Artif. Intell.*, vol. 20, no. 3, pp. 347–364, Apr. 2007.

[36] A. Salekin, M. M. Rahman, and S. H. Chowdhury, "Pattern matching in time series using combination of neural network and rule based approach," in *Proc. 7th Int. Conf. Electr. Comput. Eng.*, Dec. 2012, pp. 478–481.

[37] S. Alaee, R. Mercer, K. Kamgar, and E. Keogh, "Time series motifs discovery under DTW allows more robust discovery of conserved structure," in *Data Mining and Knowledge Discovery*, vol. 35, no. 3. New York, NY, USA: Springer, 2021.

[38] P. Senin, "Dynamic time warping algorithm review," Dept. Inf. Comput. Sci., Univ. Hawaii Manoa, Honolulu, HI, USA, Tech. Rep., 2008, pp. 1–23. [Online]. Available: https://csdl.ics.hawaii.edu/techreports/2008/08-04/08-04.pdf

[39] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. Workshop Knowl. Discovery Databases*, vol. 398, 1994, pp. 359–370.

[40] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, Sep. 2011.

[41] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[42] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.

[43] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[44] F. Bação, V. Lobo, and M. Painho, "Self-organizing maps as substitutes for K-means clustering," in *Computational Science—ICCS 2005* (Lecture Notes in Computer Science), vol. 3516, no. 3. Berlin, Germany: Springer, 2005, pp. 476–483.

TOMISLAV HLUPIĆ was born in Zagreb, Croatia, in 1986. He received the B.S. degree in computing and the M.S. degree in information and communication technology from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree.

Since 2016, he has been a business intelligence consultant in various roles on various international and domestic projects. Since 2018, he has been a Teaching Assistant and a Lecturer with Algebra University College, holding courses in the business intelligence and data engineering domains. He is the author of several papers presented at international conferences. His research interests include business intelligence, data lakes, spatio-temporal data streams, and time series applications in business environment.

DRAŽEN OREŠČANIN received the B.S. and M.S. degrees from the Faculty of Electrical Engineering and Computing, University of Zagreb, where he is currently pursuing the Ph.D. degree.

Since 2001, he has been the Founder and the CEO of Poslovna inteligencija, the leading business intelligence and data warehousing vendor in Adriatic Region. Besides that, he is the author and coauthor of more than ten papers. He is also an Active Member of TM forum alliance, contributing to the ABDR standard and data governance project. His research interests include the fields of business intelligence, data warehousing, and big data analytics.

MIRTA BARANOVIĆ (Member, IEEE) is currently a Full Professor in computer science with the Faculty of Electrical Engineering and Computing, University of Zagreb. She worked as the Vice Dean for students and education with the Faculty of Electrical Engineering and Computing, University of Zagreb. Her research interests include databases, information systems, data warehouses, data lakes, and the semantic web. She is currently a member of the Croatian Centre of Research Excellence for Data Science.