,

# ICARE: An Intuitive Context-Aware Recommender with Explanations*

Anna Dalla Vecchia[1,†], Niccolò Marastoni[1,†], Barbara Oliboni[1,†] and Elisa Quintarelli[1,*,†]

[1]Dept. of Computer Science, University of Verona, Verona, Italy

Abstract

The Intuitive Context-Aware Recommender with Explanations (ICARE) framework leverages data mining algorithms to provide contextual recommendations, together with their explanations, useful to achieve a specific and predefined goal. We apply ICARE in the healthcare scenario to infer personalized recommendations related to the activities (fitness and rest periods) a specific user should pursue or avoid in order to obtain a high value for the sleep quality score, while also considering their current context and the physical activities performed during the previous days.

**Keywords**

Explainable Recommendations, Context-awareness, Data mining

## 1. Introduction

Recommendation Systems (RS) provide suggestions related to a considered decision-making process and in particular to what item is relevant for a user on the base of his/her profile or past habits. In the healthcare scenario, a Health Recommendation System (HRS) provides recommendations in different contexts, such as medical treatment suggestions, nutrition plans, or physical activities to perform in order to reach and follow a healthy lifestyle [1].

Collecting data related to people's behaviours and well-being has become easier thanks to wearable devices; indeed, many people own them and can monitor their movements with simple apps, record their sleep quality and heartbeats, while also being surrounded by IoT devices embedded in common appliances in homes, offices and means of transport. This situation ensures a constant stream of new data that can be integrated, also with external data sources, and offers increasing opportunities for data analytics in the healthcare domain to produce useful, and often implicit, insights. Moreover, collecting data can also help people to gain more awareness of their physical health and improve their lifestyle. A very important aspect of sensor data is their intrinsically temporal nature, as sensors collect information about events

*SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy*

*Corresponding author.

†These authors contributed equally.

anna.dallavecchia@studenti.univr.it (A. Dalla Vecchia); niccolo.marastoni@univr.it (N. Marastoni); barbara.oliboni@univr.it (B. Oliboni); elisa.quintarelli@univr.it (E. Quintarelli)

0000-0001-6988-1203 (N. Marastoni); 0000-0003-4395-0893 (B. Oliboni); 0000-0001-6092-6831 (E. Quintarelli)

that happen in succession. Each event is labeled with a timestamp reporting the exact time at which it is collected, thus the temporal feature is one of the contextual dimensions that can be leveraged to obtain useful knowledge, e.g. whether events are correlated to any additional contextual feature, such as weather conditions or personal circumstances.

In this paper, we will describe ICARE (Intuitive Context-Aware Recommender with Explanations), a framework presented in [2] and able to (i) collect data from wearable devices (e.g., Fitbit), (ii) enrich data with external contextual information, (iii) analyse enriched data to discover sequential rules by correlating sequences of past events, along with their intensity, with a specified future goal, in our case ”sleeping well”, and (iv) provide explainable recommendations by means of an intuitive application. In particular, ICARE suggests which set of actions is best to take next in order to have a good night’s sleep whilst considering the user’s current context, as well as the actions that should be avoided. An aging function is introduced to facilitate up-to-date analysis and consequently adapt recommendations when frequent behaviours change; for example, fitness activities correlated with sleep quality might change during the year due to different factors, such as weather conditions or available free time.

The paper is structured as follows: Section 2 informally introduces the proposed algorithm and Section 3 describes our proposal to adopt mined sequential rules to provide both positive and negative suggestions toward a specific goal. Section 4 describes the implemented mobile App, and Section 5 summarizes the state of the art. Finally, 6 concludes the paper and outlines future work.

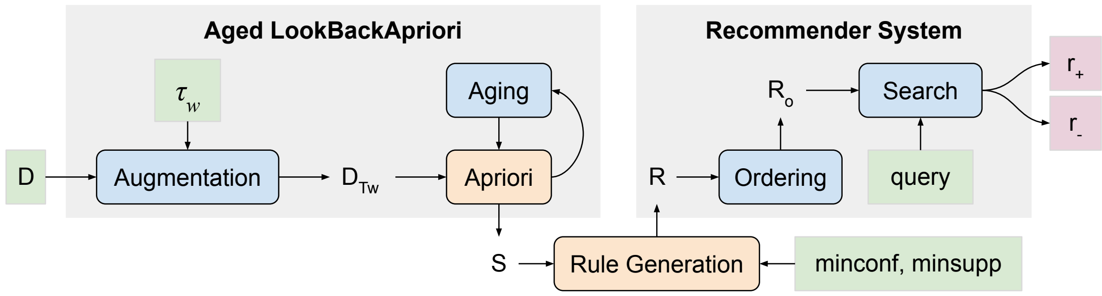



**Figure 1:** ICARE Workflow

## 2. The ALBA Algorithm

In this Section, we describe the overall approach of ICARE and the ALBA (Aged LookBack-Apriori) algorithm used to infer sequential rules that are then used to provide contextual recommendations. As shown in Figure 1, ICARE needs as input a temporal dataset $D$ (in our scenario, the log of physical activity levels collected by Fitbit), enriched with contextual information. For our scenario we leverage the temporal dimension to capture the user’s habits on some specific days, i.e., we distinguish if each day is either a weekday or part of the weekend. Whenever it is possible, we also utilize weather conditions to establish if they affect the user’s preferences. The dataset $D$ is fed into ALBA, which then considers a temporal window $\tau_w$ to con-

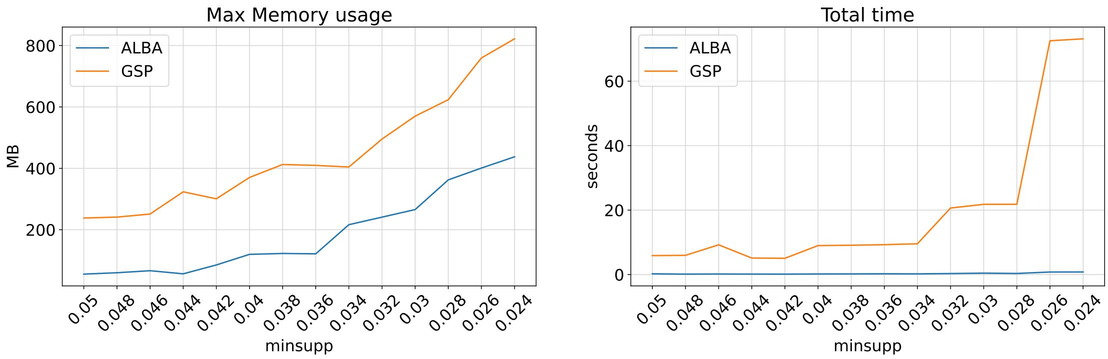



**Figure 2:** Memory usage and total time in ALBA and in GSP

struct an augmented data set $D_{\tau_w}$. This augmented dataset is then fed to a version of the Apriori algorithm [3] that calls an aging mechanism at each iteration to calculate the support of items in such a way that older items slowly decay. This process outputs a set of frequent sequences $S$, used to generate a set of totally ordered sequential rules $R$, formalized as implications $X \rightarrow Y$, where $X$ and $Y$ are two sets of ordered data items, such that $X \cap Y = \emptyset$, according to specific thresholds for confidence and support [4]. Support is the frequency of the set $X \cup Y$ in the dataset, while confidence is the conditional probability of finding $Y$, having found $X$ and is given by $\frac{sup(X \cup Y)}{sup(X)}$. Without an aging mechanism, the support of a sequence appearing frequently in recent times and another sequence that appeared the same amount of times a while ago would be the same. For this reason, we modify the support of each sequence to account for their recentness. More in detail, in order to penalize older sequences, we multiply each row of our dataset by an aging factor that guarantees that the items in the temporal window will still be represented, while older items decay but never quite disappear.

The recommender system orders $R$ with the criteria introduced in Section 3.1 to produce a set of totally ordered sequential rules that can be queried to extract a positive recommendation $r+$ and a negative recommendation $r-$. This last step is explained in Section 3.2.

**Difference with other approaches:** A sequential pattern-mining algorithm identifies frequent sub-sequences of items in the data. The algorithm typically involves two main steps: candidate generation and pattern pruning. In the candidate generation step, the algorithm generates a set of candidate patterns by exploring the search space of possible sequences. In the pruning step, the algorithm removes any pattern that does not reach the minimum support threshold, i.e., the minimum number of sequences for a pattern to be considered frequent.

We also follow this approach but with a slight difference in the candidate generation step, where the order of the itemsets is encoded in the itemsets themselves. This allows us to generate fewer candidates than usual sequence mining algorithms, resulting in less memory usage and computation time. In Fig. 2 we compare the performance of ALBA with GSP [5].

**Validation of the aging mechanism:** In order to evaluate our approach we show that our algorithm with the aging mechanism (ALBA) improves the accuracy of the version without the

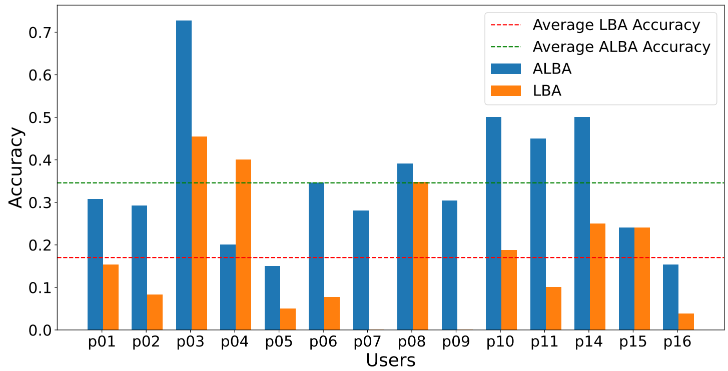



**Figure 3:** Accuracy of LBA and ALBA on PMdata users

aging (LBA). We show the evaluation on the PMData dataset [6], where ALBA outperforms LBA in 12 of the 14 users, as can be observed in Fig. 3.

## 3. The Recommender System

In our use case scenario, the transactional dataset $D$ is a log of physical activity levels and sleep quality extracted from Fitbit, where for each day $t_i$, the user's physical activities, such as heavy *activity (HA)*, *light activity (LA)*, *steps (ST)*, *rest periods (R)*, *sleeping score (SL)* and their related intensities are encoded. These values are enriched by the tag $WD$, if $t_i$ represents a day that falls on a weekend, and by additional codes relating to the weather conditions (*S=sunny*, *C=cloudy*, *R=rainy*), if the location of the user during $t_i$ is available. Weather conditions are extracted through the API of OpenWeather [7]. For each user, the time interval of each (daily) activity has been discretized into 3 possible values (*1: Low*, *2: Medium*, *3: Intense*). Each day is an itemset, as the activities themselves are not ordered due to the aggregated nature of Fitbit data.

In this dataset, the timestamp $i$ is related to an interval including the fitness activities and the subsequent sleeping period. For simplicity, we will call this time unit day. After frequent itemsets have been mined, non-contextual sequential rules will be generated and in our scenario, they will be of the form $r_i : I^f_{-(\tau_w-1)} \wedge \cdots \wedge I^f_{-2} \wedge I^f_{-1} \rightarrow I^s_0 \, [s_i, c_i]$.

This shows, with support $s_i$ and confidence $c_i$, the correlation between the sleep quality (i.e., our target function) for the current day 0 (see the itemset $I^s_0$ in the consequent related to the sleeping activity) and fitness activities, performed considering at most $\tau_w$ days, where $\tau_w$ is the considered temporal window. We remind the reader that the sequence of itemsets in the antecedent does not need to be complete. For example, a mined rule stating that after a day with medium heavy activity (*HA: 2*) and a long stretch of rest (*R: 3*), and the subsequent day with a low level of light activity (*LA: 1*), the predicted sleeping quality for the current day will likely be medium (*SL: 2*), has the form $\{HA : 2, R : 3\}_{-2} \wedge \{LA : 1\}_{-1} \rightarrow \{SL : 2\}_0 \, [s_r, c_r]$.

When the Fitbit log is enriched with contextual information, ICARE mines sequential rules of the form $r_i : CI^f_{-(\tau_w-1)} \wedge \cdots \wedge CI^f_{-2} \wedge CI^f_{-1} \rightarrow I^s_0 \ [s_i, c_i]$, where each itemset in the antecedent of the rule contains, besides the information about frequent fitness activities, also the related contextual conditions, if available. For example, a mined rule stating that after a cloudy day with a medium level of heavy activity (*HA: 2*) and a long stretch of rest (*R: 3*), and the subsequent rainy day, during a weekend, with a low level of light activity (*LA: 1*), the predicted sleeping quality for the current day will likely be medium (*SL: 2*), has the following form:

$$\{C, HA : 2, R : 3\}_{-2} \wedge \{R, WD, LA : 1\}_{-1} \rightarrow \{SL : 2\}_0 \ [s_r, c_r]$$

To discover the best rule $\bar{r}$ for predicting the answer to the query *"How well will I sleep tonight?"*, we need to match the mined rules to a portion of the user's Fitbit log

$$L = \langle I'_{-(\tau_w-1)}, \dots, I'_{-2}, I'_{-1} \rangle$$

related to the established temporal window. This partial log will be hereafter called query. Note that if the user constantly wears the smartwatch, the query will contain information for each day, always enriched with contextual information related to the day of the week ($WD$ or not), and sometimes the weather conditions, when available. For example, during the previous 3 days the user log/query may be: $L = \langle \{LA : 1, MA : 3, HA : 2, R : 3\}_{-3}, \{LA : 3, MA : 1, HA : 1, R : 3\}_{-2}, \{LA : 2, MA : 2, HA : 3, R : 2\}_{-1} \rangle$.

Note that $L$ refers to weekdays (the item $WD$ related to the weekend is not present) and the weather information is not available (i.e., the user did not declare his/her current location). The algorithm will then need to sift through all the mined rules to find one that matches the query in the antecedent and the goal in the consequent.

## 3.1. Rule Ordering

The set of rules is ordered using the following criteria: 1) confidence, 2) completeness, 3) support, and 4) size. Ordering by confidence and support is straightforward, as they are simple float values. A rule with better support will be prioritized over a rule with less support and the same goes for the confidence. On the other hand, ordering by completeness means that the rules that have at least one type of activity per timestamp in the considered temporal window will be prioritized over those that lack information in specific days. Let us define the subset of empty itemsets in a rule $r$ as $I^r_\emptyset : \{I_i \in r \mid I_i = \emptyset\}$. We can define the completeness order between two rules $r_1, r_2$ as $r_1 >_c r_2 \rightarrow |I^{r_1}_\emptyset| < |I^{r_2}_\emptyset|$ and $r_1 =_c r_2 \rightarrow |I^{r_1}_\emptyset| = |I^{r_2}_\emptyset|$.

If two rules $r_1, r_2$ have the same support, confidence and $r_1 =_c r_2$, then the rule with more itemsets will be prioritized: $r_1 >_c r_2 : |r_1| > |r_2|$.

## 3.2. Rule Search

The recommender system is designed to be goal-oriented, so the rules are first filtered (according to their consequent) into two different sets $\mathscr{R}^+(\bar{r})$ and $\mathscr{R}^-(\bar{r})$, that will contain the recommendations on the fitness activities that may lead to better sleep quality and to worse sleep quality, respectively.

The two sets of rules are then searched separately to produce a positive recommendation and its negative counterpart. Before defining how rules are matched with queries, we need to define the similarity of itemsets and items. Every item is either a contextual item or a physical activity represented by its intensity and duration, thus two items are considered similar if they have the same type, i.e., *LA:3* is similar to *LA:2* but not to *MA:3*. Note that the contextual value can match only with itself. Given an ordered set of rules, obtained through the above criteria, the algorithm needs to find the best rule that matches the query. The query is a list of activities and contextual information, if any, in sequential time slots and it is matched to the antecedent of a rule through the following criteria:

1. *EXACT_MATCH* → the query is exactly the antecedent of the rule:
   **Query:** $\{LA : 3\}_{-3} \wedge \{MA : 2\}_{-2} \wedge \{R : 3\}_{-1}$
   **Match:** $\{LA : 3\}_{-3} \wedge \{MA : 2\}_{-2} \wedge \{R : 3\}_{-1}$
2. *MATCH* → all of the items in the matching rule appear in the right time slot in the query:
   **Query:** $\{MA : 2, LA : 3\}_{-3} \wedge \{MA : 2\}_{-2} \wedge \{WD, LA : 1, R : 3\}_{-1}$
   **Match:** $\{LA : 3\}_{-3} \wedge \{MA : 2\}_{-2} \wedge \{WD, R : 3\}_{-1}$
3. *PARTIAL_MATCH* → some of the items in the query appear in the right time slot in the matching rule, while others have "similar" counterparts in the right time slot:
   **Query:** $\{R : 2, LA : 2\}_{-3} \wedge \{MA : 2\}_{-2} \wedge \{L : 2, R : 2\}_{-1}$
   **Match:** $\{LA : 3\}_{-3} \wedge \{MA : 2\}_{-2} \wedge \{R : 3\}_{-1}$
4. *SIMILAR_MATCH* → every time slot in the query contains one item that is "similar" to one item in the corresponding time slot of the matching rule:
   **Query:** $\{R : 3, LA : 2\}_{-3} \wedge \{MA : 3\}_{-2} \wedge \{L : 2, R : 1\}_{-1}$
   **Match:** $\{LA : 3\}_{-3} \wedge \{MA : 2\}_{-2} \wedge \{R : 3\}_{-1}$

The search algorithm then iterates over the ordered rules and returns the best match (or no match at all) according to the criteria above. If an exact match is encountered, the algorithm immediately returns the corresponding rule $\bar{r}$ iteration proceeds until the end and collects the other types of matching rules in their respective lists. At the end, the best rule $\bar{r}$ is the first rule in the first non-empty list in the order established above and is of the form:

$$\bar{r} : I^f_{-(\tau_w-1)} \wedge \cdots \wedge I^f_{-2} \wedge I^f_{-1} \rightarrow I^{s_0}$$

If all lists are empty, the algorithm returns NULL. This process is executed for both $\mathscr{R}^+(\bar{r})$ and $\mathscr{R}^-(\bar{r})$, thus returning the best possible positive recommendation and the best negative recommendation. The two sets are of the form:

$$\mathscr{R}^+(\bar{r}) = \{I^f_{-(\tau_w-1)} \wedge \cdots \wedge I^f_{-2} \wedge I^f_{-1} \wedge \mathbf{I^f_0} \rightarrow I^{*s}_0 \textit{ with } I^{*s}_0 > I^s_0\}$$
$$\mathscr{R}^-(\bar{r}) = \{I^f_{-(\tau_w-1)} \wedge \cdots \wedge I^f_{-2} \wedge I^f_{-1} \wedge \mathbf{I^f_0} \rightarrow I^{*s}_0 \textit{ with } I^{*s}_0 < I^s_0\}$$

The set $\mathscr{R}^+$ is composed of the rules with the same past activities of $\bar{r}$, but with a suggestion of fitness activities for the current day (i.e., $\mathbf{I^f_0}$) and with a higher sleeping quality in the consequent. On the contrary, the rules in $\mathscr{R}^-$ are those with the same past activities of $\bar{r}$, but with a suggestion of fitness activities for the current day (i.e., $\mathbf{I^f_0}$) that may lead to worse sleeping quality in the consequent. The order relation $>$ depends on the function we want to optimize. Note that the antecedent of a sequential rule is the explanation of the current suggestion, i.e., it is the recent behaviour of the user that leads to a certain sleep quality. It is important to highlight that, when using sequential rules to provide recommendations, the sequential relationship between each itemset is important. Indeed, the user performs activities on particular days and under precise contextual conditions, thus the order is important. This is the reason why we have developed our approach without starting from well know algorithms, like GSP [5].

## 4. ICARE App

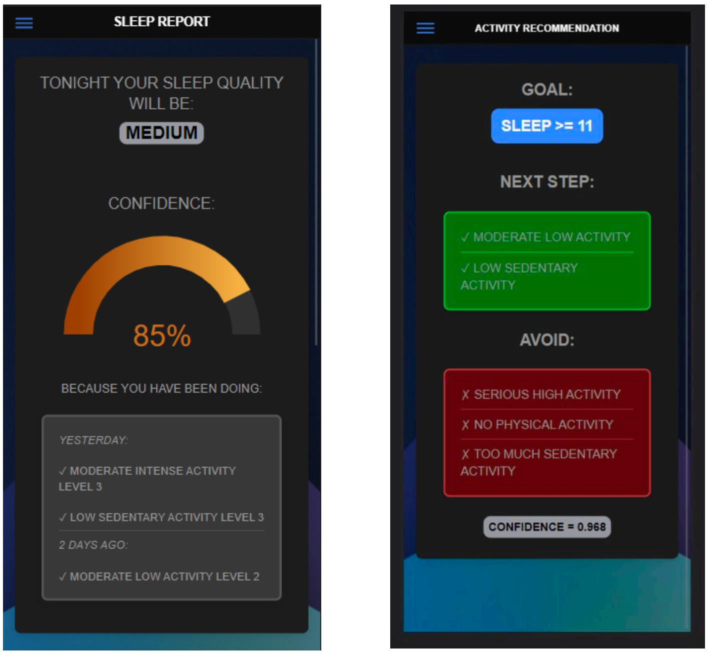
We implemented an Android app with a Python backend which recommends activities to perform during the current day, either a weekday or during the weekend, in order to increase the sleeping quality w.r.t. the predicted value, as well as the activities to avoid.

The app is written in Ionic, an open-source framework used to create hybrid mobile apps. Its main features resemble many existing health-related apps, collecting data from the user such as their weight, sleep quality, and activity levels. The latter are obtained directly from a Fitbit device worn by the user and are used mainly in the sleep quality prediction and the activity recommendation phases. The main view and the sleep quality prediction page, together with its explanation, can be seen in Figure 4(a), while positive and negative recommendations are shown in Figure 4(b). The ICARE app provides a personalized prediction by giving a sleep quality evaluation for the following night, with a visual representation of the related confidence, describing the expected likelihood that the prediction will turn out to be true. Moreover, we provide the user with an explanation for highlighting why that prediction was proposed. In particular, physical activities performed in the previous days, and considered as the prevision reason, are summarized. This way users are given an insight into the motivations that led to a prediction, and thus they can better understand how activities and desired goals are intertwined. Starting from the description of the present situation and the related prediction, ICARE provides users with personalized suggestions that are different for different users. The app shows the goal to reach, together with positive and negative recommendations describing activities to perform and to avoid, respectively, in order to achieve the given goal. Proposed suggestions change over time and are strictly related to user context.

## 5. Related Work

Data Mining algorithms considering the temporal dimension have been proposed in the literature and mainly infer sequences of events [8, 9, 10, 11]. In our proposal, we set the minimum time gap to the daily granularity of Fitbit data, the maximum time gap is flexible since the temporal window can be enlarged but we differ from other previous work because we need to maintain the relative (w.r.t. the current day) temporal information of each itemset, since it is used to provide in time recommendations and thus, the antecedent of mined rules has to match with the real user log storing what a user has done daily in the past few days.

The impact of contextual information has been investigated in the state of the art and considered relevant for providing personalized suggestions, since Context-Aware Recommendation Systems (CARS) are able to provide more accurate recommendations by adapting them to the specific context the user is acting in [12, 13]. Here, the notion of context can include, but it is not limited to, geographical, temporal, social and categorical information. For example, the




(a) (b)

**Figure 4:** (a) ICARE Prediction and Explanation (b) ICARE Positive and Negative Recommendations

ability to discover that a particular user sleeps better after some days of heavy fitness activity and that the same user prefers to train during sunny days can be useful to provide personalized suggestions suited to the weather conditions of the next few days.

In the recommendation field, the importance of explanations has been also emphasized, since they could help users fulfill their needs more easily and in an intuitive way, but also accept the suggestions [14]. This is true especially in the health care domain, since people using wearable devices do not have clinical expertise, thus the ability to provide clear and quick recommendations, together with intuitive explanations, could be very important [15].

## 6. Conclusions

The paper introduces ICARE, a framework for collecting and enriching wearable device data to produce contextual sequential rules. Sequential rules correlating sequences of past events with a specified future goal are discovered by analysing temporal data. Finally, ICARE provides explainable recommendations by means of an intuitive application. As for future work, we plan to extend the ICARE app in order to collect the user's opinion on the received predictions and recommendations, and to infer not just the frequent activity patterns, but also the average duration of each activity correlated with additional external information (e.g., the daily schedule). Indeed, it could be useful to suggest the next best activities on the base of the user's free time, whilst factoring the current time, the available remaining time interval, and the average duration of the activities to be recommended.

## References


[1] A. Calero Valdez, M. Ziefle, K. Verbert, A. Felfernig, A. Holzinger, Recommender Systems for Health Informatics: State-of-the-Art and Future Perspectives, Springer International Publishing, Cham, 2016, pp. 391–414.

[2] B. Oliboni, A. Dalla Vecchia, N. Marastoni, E. Quintarelli, ICARE: An Intuitive Context-Aware Recommender with Explanations, Springer Nature, To appear.

[3] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, p. 487–499.

[4] C. Zhang, M. Lyu, W. Gan, P. S. Yu, Totally-ordered sequential rules for utility maximization, CoRR abs/2209.13501 (2022).

[5] S. K. Harms, J. S. Deogun, Sequential association rule mining with time lags, J. Intell. Inf. Syst. 22 (2004) 7–22.

[6] V. Thambawita, S. A. Hicks, H. Borgli, H. K. Stensland, D. Jha, M. K. Svensen, S.-A. Pettersen, D. Johansen, H. D. Johansen, S. D. Pettersen, et al., Pmdata: a sports logging dataset, in: Proceedings of the 11th ACM Multimedia Systems Conference, 2020, pp. 231–236.

[7] OpenWeather, 2023. URL: https://openweathermap.org/api.

[8] R. Agrawal, R. Srikant, Mining sequential patterns, in: P. S. Yu, A. L. P. Chen (Eds.), Proceedings of the Eleventh International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan, IEEE Computer Society, 1995, pp. 3–14.

[9] S. K. Harms, J. S. Deogun, Sequential association rule mining with time lags, J. Intell. Inf. Syst. 22 (2004) 7–22.

[10] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M. Hsu, Mining sequential patterns by pattern-growth: The prefixspan approach, IEEE Trans. Knowl. Data Eng. 16 (2004) 1424–1440.

[11] M. J. Zaki, SPADE: an efficient algorithm for mining frequent sequences, Mach. Learn. 42 (2001) 31–60.

[12] G. Adomavicius, A. Tuzhilin, Context-Aware Recommender Systems, Springer US, Boston, MA, 2011, pp. 217–253. doi:10.1007/978-0-387-85820-3_7.

[13] G. Adomavicius, B. Mobasher, F. Ricci, A. Tuzhilin, Context-aware recommender systems, AI Magazine 32 (2011) 67–80. doi:10.1609/aimag.v32i3.2364.

[14] K. Balog, F. Radlinski, S. Arakelyan, Transparent, scrutable and explainable user models for personalized recommendation, SIGIR'19, Association for Computing Machinery, New York, NY, USA, 2019, p. 265–274.

[15] C. Combi, B. Amico, R. Bellazzi, A. Holzinger, J. H. Moore, M. Zitnik, J. H. Holmes, A manifesto on explainability for artificial intelligence in medicine, Artif. Intell. Medicine 133 (2022) 102423.