Mining User Behavior: Inference of Time-boxed Usage Patterns from Household Generated Data

1st Ramona Tolas
Computer Science Department
Technical University
Cluj-Napoca, Romania
ramona.tolas@cs.utcluj.ro

2nd Raluca Portase
Computer Science Department
Technical University
Cluj-Napoca, Romania
raluca.portase@cs.utcluj.ro

3nd Mihaela Dinsoreanu
Computer Science Department
Technical University
Cluj-Napoca, Romania
mihaela.dinsoreanu@cs.utcluj.ro

4rd Rodica Potolea
Computer Science Department
Technical University
Cluj-Napoca, Romania
rodica.potolea@cs.utcluj.ro

***Abstract*—The growth of technology and the reduced cost of data storage are enablers for producing and storing a large amount of data. Smart household devices are a category of data producers due to the monitoring sensors equipping the device. These sensors monitor the device's state and user interaction with the device. Besides the initial reason for planting the monitoring equipment, further valuable information can be extracted from this data, such as user behavior. Mining usage patterns can further be used in forecasting user presence, data-driven decisions, or service personalization. A processing pipeline for mining usage patterns is proposed in this paper. The problem is theoretically formulated and a method of mining usage patterns is proposed. The method is developed and tested on synthetic data and interesting insights are extracted from real data by deploying the pipeline in a data lake containing real interactions of users with smart home appliances. Similarities between real usages of household appliances are found as a result of this step and sevaral categories of users are defined based on them.**

Index Terms*—knowledge inference; clustering; event-based signal processing; pattern mining; household-generated data; usage mining

I. Introduction

The last years are considered to be one of the periods of the greatest growth of technology. This expansion of technology together with the invention of smart devices has as effect a constantly growing trend of creation and consumption of data. The reduced cost of data storage is also a powerful enabler.

A smart device is an electronic device that is usually connected with the Internet or with other devices via different communication protocols. Smart devices are equipped with sensors that measure various characteristics of both the appliance and the surroundings where the appliance is deployed. Events generated from the interaction of the user with the appliance are also captured and recorded. Most of the time, these measurements are transmitted via the Internet in data lakes [1] owned by smart appliance producers. These data lakes contain a large number of such measurements and events.

In this context, a research goal of processing this type of data and extracting useful information from it is defined in both academic and industrial worlds. Knowledge inference has many other pragmatic sub-tasks such as predictive maintenance, identification of usage patterns and user profiling.

The profile of a smart device user is a summary of the user's behavior and preferences. Mining user profiles is often used for service personalization as users differ in their interests and goals when using the same device. Mining the usage of a smart device can also be a powerful source of insights. Inference of usage patterns can be used in forecasting user presence [2]. These insights can be used further in management systems like those proposed in [2] and [3].

In this paper, we propose a formalization of the usage mining task. A processing pipeline to tackle the formalized problem is presented. The method is tested on synthetic data and deployed in production with the goal of discovering interesting real insights.

The rest of the paper is organized as follows: First, we give an overview of the related work. Section III is establishing the required theoretical background needed for formalizing the problem statement of mining user behavior in Section IV. In Section V the processing pipeline is presented and its evaluation on both synthetic and real data is described in Section VI. Conclusions, presented method limitations and future work are tackled in the last section of this work.

II. Related work

The topic of mining usage patterns is strongly represented in the literature as applied in the business domain of software services, such as web usage mining [4] [5], network usage mining [6] or API usage [7] [8]. The concept of user profiling, collecting relevant information about the user, is also frequently referring to the users of software applications [9] [10] [11].

 ISBN: 978-1-68558-082-7

However, in the current technological-driven context, the need of analysing the user has bypassed the software domain. A special type of user profiling, profiling the user presence, is presented by Barbato et al. [2], where the business domain is home automation systems. The authors tackle the topic of energy saving by proposing a system of energy management. This home automation system is proposed as part of the smart home concept. The goal is to reduce the overall energy consumption and reduce the energy peaks with intelligent management of the appliances.

The authors use context awareness as part of the management system and user presence plays an important role. For example, the heating and cooling systems, which are big consumers of energy can be pragmatically managed to function in the time frames when the user is not present in the room to avoid having multiple energy consumers. The presence monitoring is done with a system of sensors dedicated to this task - multiple infrared sensors deployed in all the rooms. The user presence is recorded and the system is forecasting future values.

Identifying user presence can also be done without planting a special monitoring system in the house, using non-intrusive systems. The smart household appliances that are already in the house and are recording the interactions of the users can be a powerful source of such insights.

Another use case of analyzing user behaviors in a non-software context is presented by Wang et al. [12]. The authors study routine in the business concept of water consumption.

Household domain and mining patterns are studied by Rahim et. al [13], where an advanced household profiling based on digital water meters are proposed. In [14], machine learning is applied to the same business concept. Gaussian Mixture Model is used to represent the water demand measurements in low dimensional feature space with the goal of pattern classification.

Data produced by smart home appliances are intensively studied by Olariu et al. [15] and Chira et al. [16], where pre-processing techniques and other associated challenges are presented in the context of data produced by smart ovens and refrigerators. Home appliance-produced data is the source of knowledge in the studies of Firte et al. [17], where data generated by washing machines and refrigerators are used for profiling their users.

As the data produced by these types of appliances is represented by large volumes, Big Data processing techniques need to be used. Portase et al. [18] present a methodology for bypassing Big Data challenges while Tolas et al. [19] is tackling transmission-related topics in the context of home appliance-generated data.

III. Theoretical Background

This section is covering basic theoretical aspects relevant to the defined goal of this paper: mining usage patterns from data produced by smart home appliances regarding user interaction.

A. Time series

The literature contains important findings and many developed libraries for pattern finding and signal processing [20] [21] [22], but for signals that are in the syntactic form of time series. This implies that the property of interest (or in this case the value of the recorded state) is measured at successive equally spaced points in time [23].

Time series are classified by the authors of [24] as the most commonly encountered data type. Given the popularity of this data type, approaches for pattern recognition applied to this type of data are in the attention of many researchers.

In order to benefit from all the research done in the time series domain, the representation of one day in the form of events unequally spaced in time needs to be transformed in time series syntactically form.

B. Taxonomy of transforming event based signal to time series

The available community and the powerful representation in the literature are making time series a preferred form of syntactical representation for the input data. However, the interaction of the user with the appliances is most of the time represented using events. Usually, the events are not equally spaced in time.

To benefit from the entire research and tooling development made in the time series domain, a transformation to this syntactical form is required. The strategy for performing this transformation is strongly connected with the business domain. Available practical mechanism of the transformation and their possible configuration and mixture are studied for realizing this taxonomy [20].

Possible methods for this transformation are:

- **Lower the frequency methods**: establishing a lower level frequency and filling the indexes which have no correspondence in the sequences of the events.
  The filling can also be done in several ways:
  - **SVP**: simple value propagation.
    The value which is propagated can also be selected:
    - **Forward-filling**: propagation of last recorded value
    - **Backward-filling**: propagation of the next recorded value
  - **ANV**: aggregation of the neighbor values (values corresponding to indexes placed in the immediate neighborhood of the index for which the value computation is made)
- **Upscaling the frequency methods**: aggregating multiple values to a higher level frequency

The frequency is dependent on the nature of the problem. The values of the observed state is also encoded in a numerical form for ease of processing.

C. Feature extraction from time series

The problem of pattern recognition in time series can be reduced to a shape-based similarity problem. Having a mechanism for determining if two time series are similar in

 ISBN: 978-1-68558-082-7

shape is a basic tool for finding patterns in time series. For computing the similarity between time series, extraction of representative features is a required step. While contributing to dimensionality reduction, this step also has a major effect on the overall performance of the data mining algorithm.

In [25] , the feature extraction methods from time-series are identified to be spread across temporal, statistical and spectral domains.

In the research [24], the authors identify several methods of determining shape similarity. One solution is using the euclidean distance, but this solution comes with associated disadvantages: sensitive to distortions and has strict requirements about the lengths of the compared time series. Dynamic Time Warping (DTW) and Longest Common SubSequence are solutions to these limitations but are computationally expensive. DTW is used in [26] to extract discriminative features and the authors report competitive results in a real-world application setup compared with other state-of-the-art methods such as InceptionTime and Convolutional Neural Network.

In [24], it is shown that shape-based similarity strategies have good results when comparing short time series. For long time series comparison, other methods such as structural similarity need to be tackled.

The shape similarity is also tackled by Zheng et al. [27], where a set of 14 shape-related features are extracted for describing financial time series. In [28], the authors use the temporal domain for feature extraction in the task of supervising artificial forest plantation trends using Google Earth Engine.

Using the frequency domain is also intensively used in the literature as a method of extracting features that describe time series.

1) Frequency domain: Projecting the signal into the frequency domain is shown to be an efficient descriptor of the time series in the literature. Schneider et al. [29] use it in the classification of cyclically recorded time series.

Nedelcu et al. [30] use the Fourier transform as a feature extraction method in the task of classifying portions of the EEG signal which are artifacts. For the same task, extracting EEG artifacts, Fast Fourier Transform and Wavelet Transform are used by Al-Fahoum et. al [31] and by Wen et al. [32].

There are also various practical implementations for frequency domain feature extraction methods such as TSFEL framework [25].

2) Discrete Fourier Transform: Discrete Fourier Transform is one method in which the frequency indicators can be included in this representation by converting a signal into individual spectral components, providing frequency information. The Fourier transform maps a signal into two vectors representing the influence of the corresponding basis function in the original signal.

A signal containing N points is represented by N complex numbers after Fourier Transform is applied. For a reduction in feature space, not all the coefficients need to be further considered. First X coefficients describe, in the form of a rough sketch, the original signal. X is determined by the business domain. In [33], an alternative strategy of selecting the coefficients which represent the signal is presented. The authors claim that selecting the largest coefficients increases the level of representation of the original signal.

D. Clustering in pattern mining

The goal of pattern mining methods is to extract interesting patterns from large data sets and use the extracted information for a better understanding of the domain or decision-making.

Clustering techniques refer to the task of partitioning a set of objects into groups with the constraints of maximizing the similarity between objects inside a group and minimizing the similarity between clusters [34]. Clustering was and still is a hot topic in computer science literature. Multiple algorithms and libraries are already developed. The authors of [34] categorize the clustering techniques into six types: partitioning (groping the objects into N groups and N is given as input parameter), hierarchical (building a dendogram), grid-based (based on space segmentation), model-based (fitting the data to a mathematical defined model) and constraint-based (clustering is based on user-defined constraints) and density-based (clusters are considered high density areas).

One advantage of density-based clustering techniques is the non-parametric approach. The number of clusters is not an input paramter of the algorithm, making it suitable for unsupervised learning models.

In a context of large databases, the research made in [35] identify DBSCAN [36], GDBSCAN [37] and DENCLUE [38] as popular density based algorithms which were developed with a focus on efficient compatibility.

Density Based Spatial Clustering of Applications with Noise (DBSCAN) is one of the most popular popular density based algorithms. The algorithm is intensively referred in the literature (at the moment of writing it has more than 27k citations in the literature). Multiple intensive used frameworks implement the DBSCAN algorithm [39], increasing the confidence of usage.

IV. Mining user behavioral patterns - Problem statement

The goal of this paper is to present a method of mining time-boxed usage patterns from data generated by smart home appliances.

A. Input data: user interactions with smart home appliances

This paper is defining a method of inferring usage profiles from data generated by home appliances capable of sensing and transmitting user interactions with the appliance.

Data generated by such appliances consist of a series of user interaction events. For the rest of this paper, the input for the defined knowledge extraction method is identified as user interaction events series (UIES), formally defined in Definition 1.

Definition 1 (User interaction events series): A user interaction event series UIES is a sequence of n events ordered in time.

$$UIES = (e_{t_1}, e_{t_2}, ...e_{t_n})$$

 ISBN: 978-1-68558-082-7

Each event is containing the timestamp of the event occurrence and the value of the observed state as defined in Definition 2. The timestamps are not necessary to be equally spaced in time.

Definition 2 (Observed state): The value of the recorded state can be one of the values of the state space S.

$$S = (S_1, S_2, ...S_m)$$

An example of input for the usage profiling method which is also used in the Experiments section of this paper is represented by the events transmitted by a smart refrigerator, having sensors that are capable of sensing when the door of the appliance is open or closed. In this case, the event is represented by the action of opening or closing the door of the smart refrigerator by the user. The recorded state is the state of the door, having the state space equal to $OPEN, CLOSED$. Another example of UIES is the interaction of the user with a smart washing machine having the capability of recording the start and end of a certain washing cycle. The events series in this case are the sequences of starting the washing cycle by the user. The observed state can be represented in this case by the type of washing cycle used by the user or the parameters of the washing cycles with which the user started the washing program.

B. Time-boxed usage representation

Part of the defined goal of this paper is to identify patterns in user behavior. This implies a certain level of recurrence of a behavior which is shifting attention to a time granularity.

Definition 3 (Time-boxed usage): Given a UIES of length n, a time interval T defined by timestamp boundaries T_{start} and T_{end} a time-boxed user interaction event series $TBES^T$ is a subset of m consecutive events of the UIES i.e.

$$TBES^T = (e_{t_{p1}}, e_{t_{p2}}, ...e_{p_m})$$

where $T_{start} \leq$t$_{pi} \leq$T$_{end}$

Definition 4 (Similar time-boxed usage): Given two time-boxed usage representations $TBES_1^T$ and $TBES_2^T$ with the same time interval T, if the distance between them is not greater than a defined threshold R, the two usage representations are similar.

Definition 5 (Time series): The time series representation of the observed state in a day is represented by sequences indexed in time with a frequency f.

$$TS.TBES = (x_{t_1}, x_{t_2}, ...x_{t_n})$$

where t_i - t_{i+1} = f

C. Behavioral pattern

Definition 6 (Behavior pattern): Given a defined time frame T, a behavior pattern is a sequence of probabilities representing the probability of the user interacting with the appliance.

$$B = (p_{t1}, p_{t2}, ...p_{tn})$$

The previously exemplified UIES, consisting of events of opening the door of a smart refrigerator, can be taken as an example. A user of a smart refrigerator could have the routine of preparing breakfast before going to work in the time interval 8 AM and 9 AM and interacting with the smart device during dinner time, between 18:30 and 20:30. This is an example of a behavioral pattern occurring every day from the week. During weekend days, the behavioral pattern might not match as the user has a different schedule. With a time granularity of one hour, the above pattern can be expressed with 24 probability values representing the probability that the user will open the door during the considered hour. The probability of the user opening the door between 8 AM and 9 AM will be close to 1 while the probability of the user opening the door between 2 AM and 1 AM will be close to zero.

D. Mining behavioral patterns - problem statement

With the defined formalism, the problem of user behavioral patterns inference is reduced to finding behavior patterns as formalized in Definition 6 given the input in form of UIES as defined in Definition 1.

V. Proposed processing pipeline for usage pattern mining

The proposed solution for inferring usage patterns from events generated by user interaction is described in Figure 1. Input in the pipeline is considered data in the syntactic form of UIES and a parameter representing the time granularity, identified by T. The syntactic form of the data is referring in this case to the data structure: events, time series, arrays of features. The T parameter is influencing the type of patterns that are extracted. For T equal to 24 hours, the effect on the processing pipeline is that daily patterns are going to be discovered. This parameter is strongly influencing the computational complexity of the overall solution as it is directly impacting the number of time-boxed usage events (TBES) which are going to be further processed.

The input is syntactically processed by a fragmentation step: user events are split into multiple time-boxed events. Following the taxonomy defined in Section III, the time-boxed events are transformed into time series, as a next step. These processes are defined as Syntactic transformations because input data is successively migrated to a different syntactic form: from UIES and TBES (events) to time-series (TS.TBES).

Applying Fast Fourier Transform [40] and selecting the first N coefficients is transposing the time-boxed events to a new representational state - spectral domain is now used to represent the user interaction events. For a given N, the number of features used to represent time-boxed events is 2*N because the Fast Fourier Transform is producing coefficients having real and imaginary parts. The time-boxed usage representation at this point is the 2*N coefficients resulting from the Transition to the spectral domain.

 ISBN: 978-1-68558-082-7

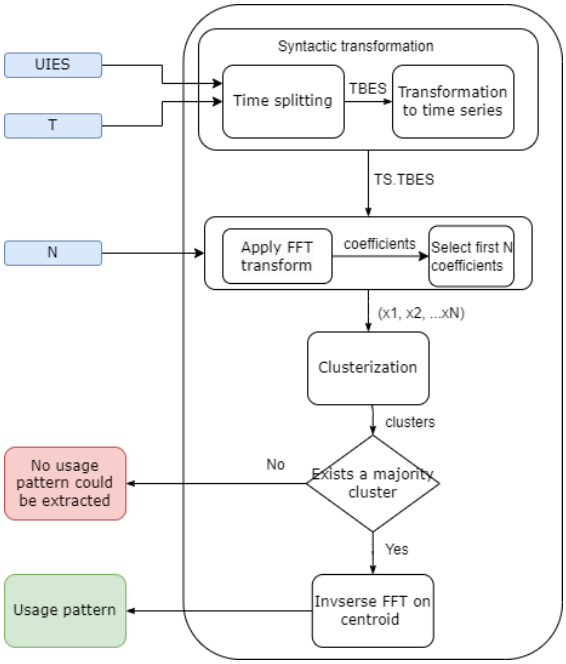

Fig. 1. Usage pattern inference pipeline

The problem of finding daily usage patterns can, at this stage, be translated to finding groups of similar usages. This step is done with clustering.

After clustering, if there is a predominant pattern in the usage of the device, the description of the usage is identified by the cluster which is holding the largest number of instances.

VI. Experiments and results

The proposed usage profiling method is tested on Door open events series, a type of UIES as mentioned in previous sections. For the rest of this paper, this type of data is referred to as Door Open Events (DOE). The time parameter for the evaluation of the processing pipeline is one day. This means that daily usage patterns are extracted from the DOE data.

A. DOE characterisation

In model development, synthetic data is utilized. The processing pipeline is also deployed in an environment containing real recordings of user interactions with refrigerator home appliances.

In TABLE I, a snapshot from a synthetic data set representing the syntactic form of DOE is shown.

TABLE I
Example of door open events from one smart refrigerator.

Timestamp	Door State
2023-01-04 08:04:35	OPEN
2023-01-04 08:05:35	CLOSED
2023-01-04 08:15:02	OPEN
2023-01-04 08:15:58	CLOSED
2023-01-04 20:11:00	OPEN
2023-01-04 20:11:35	CLOSED

For ease of processing, a numerical encoding is given to each state, as proposed in TABLE II.

TABLE II
Numerical encoding of door state

Door state	Encoding
OPEN	1
CLOSED	0

Pre-processing operations such as duplicates elimination are performed on the input data. Both the synthetically generated data and the real data are pre-preprocessed in order to reach the syntactic form described in this section (events consisting of opening and closing the door with door state encoded with numerical values of 0 and 1).

1) Synthetic data set: The synthetic data is used mainly in the development phase of the pattern mining pipeline.

A number of 6 appliances are used for the evaluation of the model. The monitored period differs from one month to years of recording. The data is configured to include certain usage patterns and noise is parametrically added to the synthetic data.

Behavioral patterns are planted in the synthetically generated data in order to provide a ground truth for the evaluation phase. In this section a briefly description of the data simulation process is provided as the focus of this paper is not the simulation strategy.

The planted behavioral patterns are based on the following concepts which are combined to obtain a behavior:

*a) **N-AP**:* N active periods in user daily interaction.

This means that in N time frames during the day the user is interacting actively with the home appliance. Outside the active period the user might also use the home appliance but with a lower frequency.

In Figure 2, a sample of DOE represented by 2-AP is shown. The snapshot contains 4 days of user interactions and there are 2 active periods.

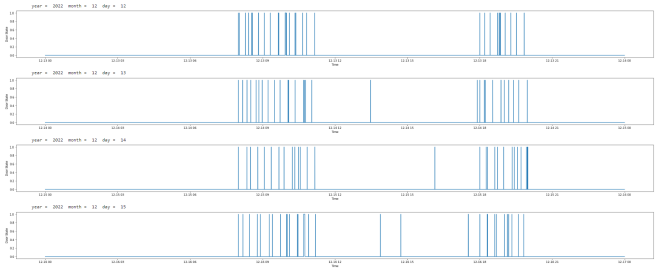

Fig. 2. Door open time series for multiple days

*b) **N-NIP**:* - This parameter represents N consecutive days with no interaction of the user with the appliance.

*c) **Noise** :* is introduced in the data for emulating real conditions. Having realistic data in the development phase is an important aspect in producing good results when the model is deployed in a real data context. The noise is introduced in the data simulation by configuring the probability to miss one active period from that day. For example, a probability of 0.1

of missing one active period for the data shown in Figure 2 means that in 1 of 10 cases, the day might not be characterized by two active periods, one of them missing.

d) ***p***$*R_{InterOpeningDurationBound}$*:* : percentage of planted random behavior. This parameter contains the number of random behaviors inserted in the data. Randomm daily behavior is represented by simulating the event of opening the door and holding the door open for a duration of time expressed in seconds and upper bounded by a threshold. The strategy is to make the duration of keeping the door open to follow the same probability distribution as the real user interactions. The number of seconds between two consecutive opening events is randomly chosen from the range [0, InterOpeningDurationBound] with equal probability for any value from the range. The InterOpeningDurationBound is simplified to IODB in the rest of the paper, hence the percentage of plated random behavior is identified by p$*R_{IODP}$.

Multiple concepts from above are combined in order to obtain the synthetic data. In TABLE III, a characterization of the appliances included in the synthetic data used for developing and evaluating the behavior mining pipeline is presented. **App id** represents the identifier of the appliance. **RP** represents the recorded period measured in days. A value of 30 for RP means that there are usage events spread over a time frame of 30 days for that appliance. **PBP** represents planted behavioral patterns and describes what behavioral models are planted in the dataset when the synthetic data is generated. For the behavior description, the parameters defined above are used. **NP** represents the noise percentage added in the synthetic data.

TABLE III
CHARACTERISATION OF THE APPLIANCES INCLUDED IN SYNTHETIC DATA

App id	RP [days]	PBP [N-AP & p$*R_{IOPB}$ & N-NIP]	NP [%]
2-AP_1	30	2AP & 0.28R_{1000}	20
3-AP_1	60	3AP & 0.28R_{1000}	20
2-AP_2	365	2AP & 0.42R_{1000}	40
1-AP_1	730	2AP & 0.42R_{1000}	20
2-AP_3	365	2AP & 0.28R_{1000} & 15-NIP	20
RB	60	Random behavior	-

2) Real data use-case - usage patterns identified in smart refrigerators: The developed processing pipeline is deployed in a data lake which contains a collection of more than 12k appliances of type smart refrigerator. The raw data is unstructured: all events generated by the interaction with the user and all recordings of the sensor deployed on the appliances generate new entry in the same general storage structure. The appliances have recorded user activity for a period which varies from one day to more than 4 years.

Pre-processing operations such as duplicates elimination, selection of events of interest, numerical encoding of door state in case of door opening events are preformed. As the pipeline is designed to process events generated from one appliance, a device selection step is performed in the real data in order to select the most meaningful devices which are feed to the processing pipeline. Analysis consisting of probability distribution of opening the door and duration of keeping the door open are used for the device selection.

The experiments are performed in a DataBricks environment [41] using Databricks Workflows (lakehouse orchestration service provided by the framework).

B. Event based signal to time series

The initial syntactic structure of the data is in the form of events, as presented in TABLE I. In Figure 3, we can see a visual representation of the events presented in the snapshot data from Figure I.

Fig. 3. Visual representation of the door open events from TABLE I

From the defined taxonomy of transforming the events into time series, a forward-filling method is used. This is practically implemented with indexation mechanisms offered by Python libraries [20].

The same data as presented in TABLE I and in Figure 3 is visually represented in Figure 4. The numerical encoding from TABLE II is used for representing the states.

Fig. 4. Door open events from Figure 3 represented as time series having sampling frequency of one second and state of the door numerically represented by 0 and 1

A resampling to one hour time granularity is made, as searched patterns are daily patterns and the same information about user interaction can be modeled with fewer data. To preserve all the interactions of the user with the home appliance the aggregation method of the resampling phase is the sum of the composing aggregates. This means that the new signal represents the number of seconds the door was open in the corresponding hour. In Figure 5, it is presented a visualization of the snapshot data after this resampling phase is done.

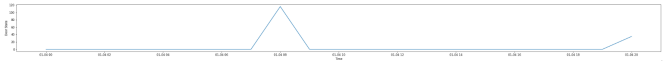

Fig. 5. Door open events from 4 after resampling the signal to hourly time granularity by summing all the seconds in which the door of the appliance was open

The result of this step is the representation of the smart refrigerator daily door state as time series indexed in time, with a frequency of one hour and value range from 0 (during that hour the door was never opened) to 3600 (the entire hour the door was open).

 ISBN: 978-1-68558-082-7

C. Feature extraction and clustering

In order to group together days with similar user behavior a clustering algorithm is used. Before applying the clustering algorithm, a feature extraction step is needed. Fast Fourier Transform is applied. After empirical research, the number of considered coefficients from the FFT method that are selected to be included in the clustering algorithm as feature is first 10 coefficients.

After this step, the events from one day of the user interaction with the appliance are represented by 20 real numbers (a coefficient from FFT has real and imaginary parts). A normalization operation is applied on all the features using a min-max scaler [42].

For clustering step, the Python implementation of the DBSCAN algorithm [43] is used, configured with euclidian distance and auto algorithm. The leaf size parameter is 30 while eps parameter and minimum samples parameter (minimum number of points to form a dense region) are empirically discovered.

Because the goal is the mining of a usage pattern which is the most frecvent behavior of the user, a cluster containing a majority of points is searched in the resulting clusters. The centroid of this cluster is the behavioral pattern.

D. User behavior pattern extraction

The centroid is used to characterize the daily usages which are grouped together. The features composing the centroid are de-normalized with the goal of using again the value space before the normalization step. Using the de-normalized centroid features, the IFFT (inverse fourier transform) is applied on those features. This step of reconstructing the time-series represents the modeling of the appliance daily usage.

In Figure 6, the usage behavior is reconstructed for the centroid of the cluster which grouped the behaviors of the data sampled in Figure 2. The pattern with 2 active periods is clearly seen from the samples and the centroid reconstruction is correctly identifying the two active periods (higher values for the hours where the user is frequently opening the door).

Fig. 6. Usage behavior reconstructed from the centroid of the cluster containing the usage represented by the sample represented in Figure 2

In order to obtain a usage behavior as described in Definition 6 the values are normalized to [0, 1] interval.

Figure 7 shows a numerical description of a usage pattern which is visually represented in Figure 8. As we can see, the active periods identified in the pattern are in the morning and in the evening.

E. Evaluation on synthetic data

Discovering the pattern is based on grouping similar days in the same cluster. As a consequence, F1-score measure is used after the clustering phase for evaluating the overall performance.

Time interval	Probability of user opening the door	Time interval	Probability of user opening the door
0-1	0.187580	12-13	0.102323
1-2	0.097206	13-14	0.160298
2-3	0.147478	14-15	0.169510
3-4	0.157930	15-16	0.134724
4-5	0.071992	16-17	0.212702
5-6	0.237944	17-18	0.135447
6-7	0.000000	18-19	0.682228
7-8	0.289808	19-20	0.897410
8-9	1.000000	20-21	0.178503
9-10	0.457622	21-22	0.086722
10-11	0.026312	22-23	0.189280
11-12	0.234447	23-24	0.078350

Fig. 7. Example of representing a behavioral pattern: the pattern is describing the interaction of a user with a smart refrigerator in 2 periods of the day

Fig. 8. Visual representation of user behavior. The user interacts with the smart appliance with a higher probability during hours 8 and 9 and also during 18 and 20.

In TABLE IV, the experiments performed for choosing the EPS parameter and MS parameter (min samples - minimum number of points to form a dense region) are described.

TABLE IV
EXPERIMENTS PERFORMED IN THE CLUSTERING PHASE WITH DBSCAN CLUSTERING ALGORITHMS

App id	EPS	MS	F1-score
2-AP_1	0.2	2	0.965
	0.75	2	0.930
	0.2	5	1.0
	0.75	5	0.904
3-AP_1	0.2	2	0.775
	0.75	2	1.0
	0.2	5	1.0
	0.75	5	1.0
2-AP_2	0.2	2	0.959
	0.75	2	0.532
	0.2	5	0.959
	0.75	5	0.566
1-AP_1	0.2	2	0.998
	0.75	2	0.596
	0.2	5	0.998
	0.75	5	0.642
2-AP_3	0.2	2	0.899
	0.75	2	0.746
	0.2	5	0.897
	0.75	5	0.833
RB	0.2	2	1.0
	0.75	2	0.666
	0.2	5	1.0
	0.75	5	0.909

F. Discovered usage behaviors in real data

As the combination of EPS = 0.2 and MS = 5 report the best performance in the synthetic data, this configuration is used when the processing pipeline is deployed in the data lake storing real data.

Data recorded from 16 devices deployed all over the world is utilized. Similar behaviors of using the smart device mainly

 ISBN: 978-1-68558-082-7

in two periods of the day are found for 37.5% of the devices. Patterns of using the appliance in mainly 3 periods of the day are found for 18.75%. Using the device in four time intervals is found in 12.5% of the analyzed devices. No behavioral pattern could be extracted from 31.25% of the devices.

VII. Conclusions And Future Work

This paper presents a processing pipeline for discovering usage patterns in the data generated by smart home appliances. The proposed method can easily be extended to any type of event-based data. A taxonomy of transforming event-based data to a more approachable syntactic form is presented.

The proposed mining method is evaluated on synthetic data which is generated with a strategy that maximizes the similarity with the real data by closely following real data characteristics. The processing pipeline is deployed in a data lake environment containing real interactions of users with smart refrigerators and interesting insights are presented.

The presented method certainly merits further investigation, especially in problems involving multiple unknowns, such as the mining of composed patterns. The experiments conducted in this study can be continued with variations in the time window which leads to mining different behaviors from the time perspective: weekly patterns, monthly patterns and even yearly patterns.

References

[1] "Data lake," https://en.wikipedia.org/wiki/Data_lake, [Online; accessed 29-March-2023].

[2] A. Barbato, L. Borsani, and A. Capone, "Home energy saving through a user profiling system based on wireless sensors," pp. 49–54, 2009.

[3] H. Abu-Bakar, L. Williams, and S. H. Hallett, "A review of household water demand management and consumption measurement," *Journal of Cleaner Production*, vol. 292, p. 125872, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652621000925

[4] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: Discovery and applications of usage patterns from web data," *Acm Sigkdd Explorations Newsletter*, vol. 1, no. 2, pp. 12–23, 2000.

[5] B. S. Kumar and K. Rukmani, "Implementation of web usage mining using apriori and fp growth algorithms," *Int. J. of Advanced networking and Applications*, vol. 1, no. 06, pp. 400–404, 2010.

[6] S.-T. Li, L.-Y. Shue, and S.-F. Lee, "Enabling customer relationship management in isp services through mining usage patterns," *Expert Systems with Applications*, vol. 30, no. 4, pp. 621–632, 2006.

[7] M. Linares-Vásquez, G. Bavota, C. Bernal-Cárdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk, "Mining energy-greedy api usage patterns in android apps: an empirical study," in *Proceedings of the 11th working conference on mining software repositories*, 2014, pp. 2–11.

[8] M. A. Saied, O. Benomar, H. Abdeen, and H. Sahraoui, "Mining multi-level api usage patterns," in *2015 IEEE 22nd international conference on software analysis, evolution, and reengineering (SANER)*. IEEE, 2015, pp. 23–32.

[9] S. Schiaffino and A. Amandi, "Intelligent user profiling," in *Artificial intelligence an international perspective*. Springer, 2009, pp. 193–216.

[10] C. I. Eke, A. A. Norman, L. Shuib, and H. F. Nweke, "A survey of user profiling: State-of-the-art, challenges, and solutions," *IEEE Access*, vol. 7, pp. 144 907–144 924, 2019.

[11] J. Peng, K.-K. R. Choo, and H. Ashman, "User profiling in intrusion detection: A review," *Journal of Network and Computer Applications*, vol. 72, pp. 14–27, 2016.

[12] J. Wang, R. Cardell-Oliver, and W. Liu, "An incremental algorithm for discovering routine behaviours from smart meter data," *Knowledge-Based Systems*, vol. 113, pp. 61–74, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095070511630332X

[13] M. S. Rahim, K. A. Nguyen, R. A. Stewart, D. Giurco, and M. Blumenstein, "Advanced household profiling using digital water meters," *Journal of Environmental Management*, vol. 288, p. 112377, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301479721004394

[14] S. McKenna, F. Fusco, and B. Eck, "Water demand pattern classification from smart meter data," *Procedia Engineering*, vol. 70, pp. 1121–1130, 2014, 12th International Conference on Computing and Control for the Water Industry, CCWI2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187770581400126X

[15] E. M. Olariu, R. Tolas, R. Portase, M. Dinsoreanu, and R. Potolea, "Modern approaches to preprocessing industrial data," in *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2020, pp. 221–226.

[16] C.-M. Chira, R. Portase, R. Tolas, C. Lemnaru, and R. Potolea, "A system for managing and processing industrial sensor data: Sms," in *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2020, pp. 213–220.

[17] C. Firte, L. Iamnitchi, R. Portase, R. Tolas, R. Potolea, M. Dinsoreanu, and C. Lemnaru, "Knowledge inference from home appliances data," in *2022 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2022.

[18] R. Portase, R. Tolas, and R. Potolea, "MEDIS: analysis methodology for data with multiple complexities," in *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2021, Volume 1: KDIR, Online Streaming, October 25-27, 2021*, R. Cucchiara, A. L. N. Fred, and J. Filipe, Eds. SCITEPRESS, 2021, pp. 191–198. [Online]. Available: https://doi.org/10.5220/0010655100003064

[19] R. Tolas, R. Portase, A. Iosif, and R. Potolea, "Periodicity detection algorithm and applications on iot data," in *2021 20th International Symposium on Parallel and Distributed Computing (ISPDC)*, 2021, pp. 81–88.

[20] "Pandas," https://pandas.pydata.org/, 2022, [Online; accessed 2-Jan-2022].

[21] "Numpy," https://numpy.org/, 2022, [Online; accessed 2-Jan-2022].

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] Wikipedia, "Time series," https://en.wikipedia.org/wiki/Time_series, [Online; accessed 24-Jan-2023].

[24] J. Lin, S. Williamson, K. Borne, and D. DeBarr, "Pattern recognition in time series," *Advances in Machine Learning and Data Mining for Astronomy*, vol. 1, no. 617-645, p. 3, 2012.

[25] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa, "Tsfel: Time series feature extraction library," *SoftwareX*, vol. 11, p. 100456, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711020300017

[26] W. Nikolai, T. SCHLEGL, and J. DEUSE, "Feature extraction for time series classification using univariate descriptive statistics and dynamic time warping in a manufacturing environment," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. IEEE, 2021, pp. 762–768.

[27] Y. Zheng, Y.-W. Si, and R. Wong, "Feature extraction for chart pattern classification in financial time series," *Knowledge and Information Systems*, vol. 63, no. 7, pp. 1807–1848, 2021.

[28] H. Fu, W. Zhao, Q. Zhan, M. Yang, D. Xiong, and D. Yu, "Temporal information extraction for afforestation in the middle section of the yarlung zangbo river using time-series landsat images based on google earth engine," *Remote Sensing*, vol. 13, no. 23, p. 4785, 2021.

[29] T. Schneider, N. Helwig, and A. Schütze, "Automatic feature extraction and selection for classification of cyclical time series data," *tm-Technisches Messen*, vol. 84, no. 3, pp. 198–206, 2017.

[30] E. Nedelcu, R. Portase, R. Tolas, R. Muresan, M. Dinsoreanu, and R. Potolea, "Artifact detection in eeg using machine learning," in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2017, pp. 77–83.

[31] A. S. Al-Fahoum and A. A. Al-Fraihat, "Methods of eeg signal features extraction using linear analysis in frequency and time-frequency domains," *International Scholarly Research Notices*, vol. 2014, 2014.

 ISBN: 978-1-68558-082-7

[32] T. Wen and Z. Zhang, “Effective and extensible feature extraction method using genetic algorithm-based frequency-domain feature search for epileptic eeg multiclassification,” *Medicine*, vol. 96, no. 19, 2017.
[33] F. Mörchen, “Time series feature extraction for data mining using dwt and dft,” 2003.
[34] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, “Dbscan: Past, present and future,” in *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, 2014, pp. 232–238.
[35] R. J. G. B. Campello, P. Kröger, J. Sander, and A. Zimek, “Density-based clustering,” *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 2, p. e1343, 2020. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1343
[36] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD’96. AAAI Press, 1996, p. 226–231.
[37] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, “Density-based clustering in spatial databases: The algorithm gdbscan and its applications.” *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, 1998. [Online]. Available: https://link.springer.com/article/10.1023/A:1009745219419
[38] A. Hinneburg and D. A. Keim, “An efficient approach to clustering in large multimedia databases with noise,” in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, ser. KDD’98. AAAI Press, 1998, p. 58–65.
[39] Scikit-learn, “DBSCAN,” https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html, [Online; accessed 19-Jan-2023].
[40] Wikipedia, “Fast fourier transform,” https://en.wikipedia.org/wiki/Fast_Fourier_transform, [Online; accessed 29-March-2023].
[41] “Databricks,” https://www.databricks.com/, [Online; accessed 29-March-2023].
[42] “Scikit-learn minmaxscaler,” https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html, [Online; accessed 29-March-2023].
[43] “Scikit-learn dbscan,” https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html, [Online; accessed 19-July-2022].

 ISBN: 978-1-68558-082-7