SMILE: a feature-based temporal abstraction framework for event-interval sequence classification

Jonathan Rebane¹ · Isak Karlsson¹ · Leon Bornemann² ·
Panagiotis Papapetrou¹

Received: 2 July 2019 / Accepted: 20 October 2020 / Published online: 23 November 2020
© The Author(s) 2020

Abstract

In this paper, we study the problem of classification of sequences of temporal intervals. Our main contribution is a novel framework, which we call *SMILE*, for extracting relevant features from interval sequences to construct classifiers. *SMILE* introduces the notion of utilizing random temporal abstraction features, we define as *e-lets*, as a means to capture information pertaining to class-discriminatory events which occur across the span of complete interval sequences. Our empirical evaluation is applied to a wide array of benchmark data sets and fourteen novel datasets for adverse drug event detection. We demonstrate how the introduction of simple sequential features, followed by progressively more complex features each improve classification performance. Importantly, this investigation demonstrates that *SMILE* significantly improves AUC performance over the current state-of-the-art. The investigation also reveals that the selection of underlying classification algorithm is important to achieve superior predictive performance, and how the number of features influences the performance of our framework.

Keywords Event interval sequences · Temporal intervals · Temporal abstractions · Classification

Responsible editor: Toon Calders.

✉ Jonathan Rebane
jonathan@dsv.su.se

Isak Karlsson
isak-kar@dsv.su.se

Leon Bornemann
leon.bornemann@hpi.de

Panagiotis Papapetrou
panagiotis@dsv.su.se

¹ Stockholm University, Stockholm, Sweden

² Hasso Plattner Institute for Software Systems Engineering, Potsdam, Germany

1 Introduction

Sequences of temporal intervals are defined as ordered sets of events occurring over time, with each event having a time duration, which may co-occur with other events. As a result, several temporal relations are possible between pairs of events, such as one event *overlapping* another event or two events starting concurrently with one *ending before* the other. Such sequences, also known as *e-sequences*, can be found in a variety of application domains, including sign language transcription (Papapetrou et al. 2009), human activity recognition and monitoring (Uddin and Uddiny 2015), music classification (Pachet et al. 1996), and predicting clinical outcomes from medical records (Kosara and Miksch 2001; Moskovitch and Shahar 2015a).

An example of an e-sequence, taken from the healthcare domain, is depicted in Fig. 1. The example e-sequence contains six events describing an Adverse Drug Reaction (ADR) caused by the use of the medication “procainamide” on a patient suffering from arrhythmia. We observe that the patient shown in the example underwent an episode of arrhythmia (first event) before being hospitalized (second event) and administered with procainamide (third event). A second episode of arrhythmia occurred shortly after (fourth event) and another dosage of procainamide (fifth event) was provided to the patient. Eventually the patient developed ventricular tachycardia (last event) which is an ADR in relation to procainamide.

Earlier work in the area of e-sequence classification has mostly focused on distance-based and feature-based classifiers.

For the case of distance-based classifiers, two state-of-the-art distance measures have been developed, i.e., Artemis (Kostakis et al. 2011) and IBSM (Kotsifakos et al. 2013). The first measure quantifies the distance between two e-sequences by measuring the fraction of temporal relations shared between them using a bipartite graph mapping, while ignoring the time duration of the individual events. On the contrary, the second measure performs a mapping of the e-sequences to vectors, with each time point described by a binary vector indicating the active and non-active events at that time point. Despite the promising classification results obtained by both measures when used in a k-NN formulation, this family of classifiers is hampered by the fact that only global properties are exploited as classification features, while local temporal properties are ignored, potentially leading to detrimental effects in predictive performance. Moreover, an extension of IBSM has been proposed for subsequence

Fig. 1 Example of an e-sequence describing an Adverse Drug Reaction (ADR) of ventricular tachycardia following the administration of procainamide to a patient suffering from arrhythmia. The e-sequence consists of six event intervals, each corresponding to a medical event, and the x-axis corresponds to the hours past since the start of the first arrhythmia event

matching in event-interval sequences, with the distance measure called ABIDE. This distance measure is used by the proposed framework in this paper.

In regard to the case of feature-based classifiers for e-sequences, the most conventional solution is to extract patterns of temporal interval relation pairs, defined based on Allen's temporal logic (Allen 1983), and use them as potential classification features (Bornemann et al. 2016) along with additional static features. This idea falls within the concept of temporal abstractions of multi-variate time series, where the main objective is to map each time series channel to an interval and then employ pattern extraction methods, such as the Karma–Lego framework (Moskovitch and Shahar 2015a) or its follow-up variants (Moskovitch and Shahar 2015b; Moskovitch et al. 2015; Batal et al. 2013; Patel et al. 2008; Karlsson and Boström 2016). The latter are, however, not direct competitors for our problem, as their target data space is multi-variate time series and not event-interval sequences. The main drawback of existing feature-based classifiers is that the temporal abstractions they employ only consider the relation types between the involved event intervals, while ignoring their actual time duration. This can impose a great deficiency in application domains where duration matters. For example, in healthcare, the duration of two overlapping medications could have a detrimental effect on the probability of the occurrence of an adverse drug event.

In this paper, we address the deficiencies of both distance-based and feature-based classifiers, by (1) considering both global and local class-predictive features, (2) taking into account both the event relation types in these discriminant features as well as their time duration.

1.1 Example

We illustrate the aforementioned deficiency with a simple example. Consider the five e-sequences depicted in Fig. 2, with event labels defined from a given alphabet $\Sigma = \{A, B, C\}$. Assume that the sequences are classified as either positive or negative. That is, S_1, S_3, S_5 are positive examples and S_2, S_4 are negative examples. Let us now consider a simple temporal pattern A **followed by** C . Note that with the term *temporal pattern* we refer to any combination of event labels described by their pairwise temporal relations. Observe that this pattern occurs in all five sequences. However, for the positive class both event intervals A and C , have a shorter time duration than those present in the negative class. Hence, any feature-based method that only considers the relation type between the intervals, ignoring their time duration, will be unable to identify this class-separation power of A **followed by** C .

Consider now a more descriptive representation of the same pattern that contains the event labels, as well as their start and end times, i.e., $\mathcal{P} = ((A, 0, 1), (C, 2, 3))$. We refer to this representation as *e-let*. Using this representation, we can capture both the temporal relation between A and C as well as the time duration of each interval. Hence, if we compute the similarity of $\mathcal{P}$ in terms of type of relation (in our case **followed by**) against all sequences, by counting the number of times this relation occurs, we can easily observe that the similarity score of $\mathcal{P}$ is the same for all five sequences; $\mathcal{P}$ occurs once in all of them resulting in an information gain of 0. On the other hand, if we also consider the time duration of the intervals, we can see that $\mathcal{P}$

Fig. 2 A database of $\mathcal{D}$ of 5 interval sequences of max length 8 with alphabet $\Sigma = \{A, B, C\}$. Sequences S_1, S_3, S_5 are classified as “-”, while S_2, S_4 are classified as “+”. An example of a class-predictive *e-let* is highlighted in red. We observe that relation *A followed by C* occurs in all five sequences, however, with different time duration. The indicated *e-let* distinguishes the positive class (“+”) from the negative class (“-”) as it only occurs in sequences S_2 and S_4 (Color figure online)

has a higher similarity to sequences S_2 and S_4 (due to the shorter time duration of *A* and *C*) compared to S_1, S_3 and S_5 . Assuming the similarity function used in the latter case is *ABIDE* (Kostakis and Papapetrou 2017), for each sequence we obtain the following similarity scores: $S_2 = 1, S_4 = 1, S_1 = 0.67, S_3 = 0.33, S_5 = 0.33$. These scores yield 4 possible attribute split points with, e.g. a decision tree classifier, out of which only $\frac{1+0.67}{2}$ separates the classes. However, the separation has the highest information gain, i.e., 1. Conversely, employing a similarity measure that takes into account both the relation type as well as the time duration of the event intervals is capable of identifying temporal patterns with higher class-separation power.

1.2 Contributions

The main contributions of our paper include a novel framework for event-interval sequence classification, a novel concept defining temporal features for this task, and a thorough empirical evaluation on 20 real-world datasets. More concretely:

- We introduce a novel, generalized framework, called *SMILE*, building upon the *STIFE* framework introduced by Bornemann et al. (2016), for classification of

event-interval sequences. The key novelty of *SMILE* is that it reduces the complexity of event-interval sequences by performing *four levels of temporal abstraction*, starting from simple, global abstraction features and gradually moving to more complex local class-predictive temporal features. These features take into consideration both the temporal relation types between the event-intervals, as well as their time duration.

- We introduce and define a new *concept* for temporal interval sequence classification, which we refer to as *e-lets*. This primitive concept describes class-predictive subsequences of interval-based events, and it is one of the key abstraction features for our framework.
- We present an extensive experimental evaluation of the proposed framework using 4 different classification models on 6 commonly used benchmark datasets, as well as on 14 datasets from the medical domain corresponding to electronic patient records of adverse drug reactions. The proposed framework achieves statistically significantly improved performance in terms of AUC over its competitors.

The remainder of this paper is organized as follows: in Sect. 2 we present the related work in the area of temporal interval sequence classification, while in Sect. 3 we formalize the classification problem studied in this paper along with the required technical background and definitions. Moreover, in Sect. 4, we present *SMILE*, the proposed framework of this paper. In Sect. 5, we provide and discuss the experimental evaluation and findings, while Sect. 6 concludes the paper and introduces directions for future work.

2 Related work

Research into sequences of temporal intervals have attracted attention within the areas of data mining and databases with original motivations focusing on simplifying complex temporal data while minimizing the loss of information. Earlier work, such as Lin (2003) have demonstrated a method to mine maximal frequent intervals, yet in the process, information loss increases as the different dimensions of the intervals become discarded. Another form of simplification commonly demonstrated is the direct mapping of sequences of temporal intervals to temporally ordered events; however such a simplification does not consider the actual duration of the intervals as seen in Giannotti et al. (2006). Various Apriori-based techniques such as from Höppner and Klawonn (2001), Mooney and Roddick (2004), and Laxman et al. (2007) exist for the discovery of temporal patterns, episodes, and association rules on interval-based event sequences. In addition, various candidate generation techniques employ approaches to reduce the exponential complexity of the mining problem, such as from Winarko and Roddick (2007) and Papapetrou et al. (2005, 2009).

Recent similarity measures pertaining to sequences of temporal intervals have been used as tools in this data domain for similarity searching, clustering and k-NN classification. As mentioned, for the k-NN family of classifiers, two state-of-the-art distance measures have been developed, one being *Artemis* introduced by Kostakis et al. (2011), and the other being *IBSM*, introduced by Kotsifakos et al. (2013). More

recently a state-of-the-art e-sequence domain similarity search framework known as ABIDE has been introduced by Kostakis and Papapetrou (2017). ABIDE is capable of generating an accurate similarity search in sequences of temporal intervals with no false dismissals alongside a relatively low computational cost. This is achieved by combining lower bounds with early abandoning methods. Importantly, the use of ABIDE over Artemis as a more informative measure should be considered as ABIDE takes into account both absolute values of the interval durations and time between intervals while Artemis does not. ABIDE should also be considered as a measure over IBSM as the later may result in false dismissals.

Building upon the foundations of these previous works, a classification framework known as STIFE, has been introduced in Bornemann et al. (2016), achieving state-of-the-art performance regarding temporal interval sequence classification. STIFE exploits such sequences in a variety of manners, employing static features, class-defined medoid features, as well as class-distinctive temporal relation pairs. However, the main limitations of STIFE is the inability to exploit information regarding event durations and relational information over a broad range of event types, as its temporal features only consider the occurrence of single event pair relations. Hence, potentially class-discriminatory temporal arrangements of events are ignored, such as those in combination across the entire span of event labels and in which the duration of such events may be highly relevant for the classification task at hand. In a different line of research targeting classification of multi-variate time series using temporal abstractions, the Karma-Lego framework (Moskovitch and Shahar 2015a) and similar variants (Moskovitch and Shahar 2015b; Moskovitch et al. 2015; Batal et al. 2013; Patel et al. 2008) are employed on healthcare temporal measurements. The key objective of Karma-Lego and its variants is the efficient discovery of frequent patterns of interval-based events of any size, which can then be employed as features for any off-the-shelf predictive model. The temporal features are constructed by enumerating the set of possible pair-wise temporal relations that may occur between the event labels contained in the training set, using the seven types of temporal relations defined in Allen's temporal logic (Allen 1983). If the e-sequences in the training set are defined over m possible event labels, in order to account for all possible relations between event labels, then the feature space comprises $7 \cdot \frac{m(m-1)}{2}$ features. The corresponding feature values are determined by the number of occurrences of each temporal relation in a given e-sequence.

Despite the competitive predictive performance of these approaches, their main drawback is the fact that they only consider the types of temporal relations occurring between the involved events, while ignoring the actual duration of these relations. In many practical scenarios, such as in healthcare, the duration of, e.g., an overlap, or a gap, may convey critical value in terms of acting as a class-discriminant feature and predicting a future event of interest.

In addition, there exist alternative approaches that extract sequential patterns as a means of building meaningful temporal features for classifiers. Such methods include SPAM (Ayres et al. 2002), for mining tradition sequential patterns, CloSpan for mining closed sequential patterns (Yan et al. 2003), GoKrimp for mining compressing sequential patterns (Lam et al. 2014), and SCIP for building classifiers based on mined

interesting patterns (Zhou et al. 2015). As such, we consider such approaches competitors to exploiting sequences of temporal intervals.

3 Problem setting

The problem studied in this paper is the classification of sequences of temporal intervals. In this section, we introduce the problem by providing the necessary definitions followed by the problem formulation.

Let $\Sigma = \{e_1, \dots, e_m\}$ be an alphabet of m event labels. An *event-interval* is an event that occurs over a time interval, while an ordered multi-set of event-intervals defines an event-interval sequence. Next, we define these two terms more formally.

Definition 1 (*Event-interval*) An *event-interval* is formally defined as a triplet $S = \langle e, t_s, t_e \rangle$, where $S.e \in \Sigma$ is the event label for that time interval, and $S.t_s, S.t_e$ correspond to the start and end times of S , respectively. Naturally it holds that $S.t_s \leq S.t_e$, where the equality is satisfied when the event is instantaneous.

We say that an event-interval $S = \langle e, t_s, t_e \rangle$ is *active* during its defined time span, i.e., from t_s to t_e .

Definition 2 (*e-sequence*) A sequence of event-intervals, also known as *event-interval sequence*, or *e-sequence*, denoted as $\mathcal{S} = \{S_1, \dots, S_n\}$, is an ordered multi-set of n event-intervals. The temporal order of the event-intervals in $\mathcal{S}$ is ascending based on their start time and in the case of ties it is descending based on their end time. If ties still exist, the event-intervals are sorted alphabetically.

The *length* of an e-sequence $\mathcal{S}$ is defined as the time-span of the e-sequence, i.e., $|\mathcal{S}| = S_n.t_e - S_1.t_s$, while the *size* of $\mathcal{S}$ is the number of event-intervals in the e-sequence. For example, the e-sequence depicted in Fig. 1 is of length 10 and its size is 6.

Let $\mathcal{D} = \{\mathcal{S}_1, \dots, \mathcal{S}_N\}$ define an *e-sequence dataset*, i.e., a collection of e-sequences. Moreover, let us assume that each e-sequence $\mathcal{S}_i \in \mathcal{D}$ is assigned with a class label $c_i \in \mathcal{C}$, with $\mathcal{C}$ being a predefined set of class labels. Hence, let $\mathcal{X} = \{\{\mathcal{S}_1, c_1\}, \dots, \{\mathcal{S}_N, c_N\}\}$ denote a *labelled e-sequence dataset* that is defined over $\mathcal{D}$ and $\mathcal{C}$.

The problem studied in this paper is to learn a classification model f from $\mathcal{X}$ that can correctly assign a new (previously unseen) e-sequence $\mathcal{S}$ with a class label from $\mathcal{C}$.

Problem 1 (*e-sequence classification*) Given a labeled e-sequence dataset $\mathcal{X}$, with each e-sequence assigned with a class label from $\mathcal{C}$, we want to learn a mapping function $f_{\mathcal{X}, \mathcal{C}} : \mathcal{S} \rightarrow \mathcal{C}$ defined over $\mathcal{X}$ and $\mathcal{C}$, with $\mathcal{S} \in \mathcal{X}$, such that for an independent labeled dataset of previously unseen e-sequences $\mathcal{X}'$, the expectation of the classification function loss function $E_{(\mathcal{S}_i, c_i) \in \mathcal{X}'}[\mathcal{L}(c_i, f(\mathcal{S}_i))]$ is minimized. The classification loss function $\mathcal{L}$ is defined as follows:

$$\mathcal{L}_{\mathcal{X}'}(c_i, f(\mathcal{S}_i)) = \begin{cases} 0 & \text{if } f(\mathcal{S}_i) = c_i, \text{ with } \mathcal{S}_i \in \mathcal{X}', \\ 1 & \text{otherwise.} \end{cases}$$

4 SMILE: a generalized temporal abstraction framework for classifying sequences of temporal intervals

We introduce SMILE, a generalized framework for classification of sequences of temporal intervals, that performs four levels of temporal abstraction. Towards this end, the following four types of abstraction features are considered, the first three of which having also been employed by the STIFE framework (Bornemann et al. 2016):

- **Static** corresponding to a set of static features providing global, aggregate statistics of an e-sequence (Sect. 4.1).
- **Medoids** corresponding to a set of similarity values of an e-sequence compared to class-medoid e-sequences from the training set (Sect. 4.2).
- **Interval relation pairs** corresponding to pairs of temporal intervals covering all combinations of pair-wise temporal relations (as defined by Allen’s temporal logic Allen (1983)) between all pairs of event labels in the given alphabet Σ (Sect. 4.3).
- **E-lets** corresponding to class-predictive subsequences of event-intervals, called e-lets with high utility (Sect. 4.4).

Next, we introduce the above feature types starting from simple static metrics and progressing to more complex temporal abstractions, exploiting both global and local temporal information from the labeled training dataset of e-sequences.

4.1 Static features

For a given e-sequence $\mathcal{S} = \{S_1, \dots, S_n\}$, as defined in Sect. 3, we consider 14 static features. For each feature described below, we also compute its corresponding value using the example e-sequence depicted in Fig. 1.

- (i) **Duration** the e-sequence length $|\mathcal{S}|$; in our example, $\text{duration}(\mathcal{S}) = 10$.
- (ii) **Size** the total number of event-intervals in $\mathcal{S}$; in our example, $\text{size}(\mathcal{S}) = 6$.
- (iii) **Dim_count** the number of unique event labels in $\mathcal{S}$, i.e.,

$$\text{dim_count}(\mathcal{S}) = |\{S.e \mid S \in \mathcal{S}\}|,$$

with $\text{dim_count}(\mathcal{S}) = 4$.

- (iv) **Start** the start time of the first event-interval in $\mathcal{S}$, i.e., $S_1.t_s$; in our example, $\text{start}(\mathcal{S}) = 0$.
- (v) **Majority** the alphabet label $e^* \in \Sigma$ that has the highest occurrence frequency in $\mathcal{S}$; ties are resolved randomly. That is

$$e^* = \text{max_freq}(\mathcal{S}) = \arg \max_{\Sigma} \sum_{i=1}^n \mathbf{1}(\Sigma_j = S_i.e).$$

In our example, $e^* = \text{'arrhythmia'}$.

(vi) **Density** the sum of interval duration values in $\mathcal{S}$, i.e.,

$$\text{density}(\mathcal{S}) = \sum_{i=1}^n \{S_i.t_e - S_i.t_s\},$$

with $\text{density}(\mathcal{S}) = 20$, for our example.

(vii) **μ Density** the mean density of event-intervals in $\mathcal{S}$, i.e., $\bar{\text{density}}(\mathcal{S}) = 3.33$.

(viii) **Concurrency** the maximum number of event-intervals that are concurrently active in $\mathcal{S}$. Let $\mathcal{V}(\mathcal{S}) = \{V_1, \dots, V_{|\mathcal{S}|}\}$ be the binary representation of $\mathcal{S}$, where each V_j is an n -dimensional binary vector, with n being the number of event-intervals in $\mathcal{S}$. Moreover, $V_j[i] = 1$ if event-interval $S_i \in \mathcal{S}$ is active at time point j and $V_j[i] = 0$, otherwise (similar to the formulation in Kotsifakos et al. (2013)). Then, the maximum number of concurrently active event-intervals in $\mathcal{S}$ is

$$\text{concurrent}(\mathcal{S})^* = \max_{j=1}^{|\mathcal{S}|} \sum_{i=1}^n V_j[i]$$

In our example, $\text{concurrent}(\mathcal{S})^* = 3$.

(ix) **Max_concurrency** the time duration of the period with the highest number of concurrent intervals in $\mathcal{S}$, i.e., $\text{concurrent_dur}(\mathcal{S}) = 1$.

(x) **μ Concurrency** the maximum concurrent interval duration normalized by *length*, i.e., $\bar{\text{concurrent_dur}}(\mathcal{S}) = 0.1$.

(xi) **Pause_time** the total duration in $\mathcal{S}$ with no active event-interval, i.e., $\text{pause_time}(\mathcal{S}) = 0$.

(xii) **μ Pause_time** the pause time normalized by *length*, $\mu\text{pause_time}(\mathcal{S}) = 0$.

(xiii) **Activity** the total duration with at least one active event-interval, i.e., the inverse of *pause time*. In our example, $\text{activity}(\mathcal{S}) = 10$.

(xiv) **μ Activity** the active time normalized by *length*, i.e., $\mu\text{activity}(\mathcal{S}) = 1$.

Complexity We observe that the above set of features consists of static summarization metrics of an e-sequence, while requiring low computational runtime. In fact, after sorting the event-intervals of an e-sequence (i.e., $\Theta(n \cdot \log(n))$) all metrics can be calculated in either $\Theta(1)$ or $\Theta(n)$. Thus, the overall runtime complexity of extracting static features from an e-sequence is $\Theta(n \cdot n \cdot \log(n))$, while only requiring $\Theta(n)$ additional memory, since the number of static features is constant. It follows that the time required to extract static features for an unseen sequence $\mathcal{S}_{\text{new}}$ is $\Theta(n \cdot \log(n))$.

4.2 Class-based medoid distance features

The main idea behind this type of e-sequence summarization approach is to extract a set of representative e-sequences from the training set and use them to map the e-sequences to a vector space. More concretely, for each set of e-sequences belonging to a class label, we compute their corresponding within class medoid using the IBSM distance function as defined in Bornemann et al. (2016). Hence, for a k -class classi-

fication problem, we extract a set $\mathcal{M} = \{M_1, \dots, M_k\}$ of k medoids functioning as representatives. We assume that each medoid $M_j \in \mathcal{D}$.

Then, given a dataset $\mathcal{D} = \{S_1, \dots, S_N\}$, each $S_i \in \mathcal{D}$ is mapped to a k -dimensional vector defined as

$$\mathcal{C}_{\mathcal{M}}(S_i) = \{IBSM(S_i, M_1), \dots, IBSM(S_i, M_k)\}.$$

For each S_i , the resulting k -dimensional vector acts as a set of k features that are passed over to the classifier.

Complexity Since the class labels (and thus cluster labels) are given, the clustering takes $\Theta(N)$ time. Afterwards we need to calculate the medoid of each cluster and subsequently calculate the distance to those for all training sequences. Assuming the number k of classes is constant we know that the size of each cluster can be at most $\Theta(N)$. For each cluster all compressed event tables (produced by IBSM) and their pairwise distances ($\Theta(N^2)$) need to be computed and stored. Thus, the runtime and memory complexity of finding the distances to all class-cluster medoids is $\Theta(N^2 \cdot m \cdot (|\Sigma| + \log(m)))$ time and $\Theta(N^2 \cdot m \cdot |\Sigma|)$ memory. The online feature extraction requires $\Theta(m \cdot (|\Sigma| + \log(m)))$ time and $\Theta(m \cdot |\Sigma|)$ memory.

4.3 Interval relation-pair features (2-lets)

This set of features involves temporal relation features between pairs of event intervals, also referred to as 2-shapelets (Bornemann et al. 2016) or 2-lets. In simple terms this stage of the framework incorporates e-sequence features corresponding to temporal relations across all possible event label pairs in $\mathcal{D}$. We consider seven types of temporal arrangements, as defined in Allen (1983) and depicted in Figs. 1 and 3.

More concretely, let A and B be two event intervals with the following property: $A.t_{start} \leq B.t_{start}$ (B does not start before A). We define the set of possible temporal relations as $\mathcal{R} = \{\text{meets, matches, overlaps with, left-contains, contains, right-contains, is followed by}\}$ that can define the temporal arrangement of A and B . The individual definitions of these relations are visualized in Figs. 1 and 3 and can be found in Papapetrou et al. (2009). We denote as $rel(A, B) \in \mathcal{R}$ the temporal relation between A and B .

These temporal relations for event intervals have already been used in the context of distance measures for e-sequences on multiple occasions such as by Kotsifakos et al. (2013) and Kostakis et al. (2011). Note that for an ordered pair of event intervals exactly one of these relations applies, meaning the temporal relation of two event intervals is unambiguous. Based on this, a 2-let can be defined as follows.

Definition 3 (2-let) Given two alphabet labels $e_i, e_j \in \Sigma$, a 2-let is defined as $l_2 = (e_i, e_j, r)$, where $r \in \mathcal{R}$ is the temporal relation between the e_i and e_j .

Given an e-sequence $\mathcal{S}$ and a 2-let $l_2 = (e_i, e_j, r)$, we say that l_2 occurs in $\mathcal{S}$ if there exists at least two event intervals S_k and S_l in $\mathcal{S}$, such that $S_k.e = e_i$, $S_l.e = e_j$, and $rel(S_k, S_l) = r$.

All 2-lets of an e-sequence $\mathcal{S}$ can be found by simply determining the relations of all pairs of event-intervals (A, B) , where $A, B \in \mathcal{S}$ and B does not occur before A . The

Algorithm 1: Extract and select all 2-let features

Require: Let $\mathcal{D} = \{S_1, \dots, S_N\}$ be the set of e-sequences, Σ be the event alphabet and k the number of features to keep.

```

for  $i \leftarrow 1$  to  $n$  do
  for  $(A, B) \in \{(A, B) \mid A, B \in S \wedge A.t_{start} - \epsilon \leq B.t_{start}\}$  do
     $r \leftarrow$  temporal relationship of  $(A, B)$ 
     $j \leftarrow$  compute column index of e-let  $(A, B, r)$ 
     $M^{(i,j)} \leftarrow M^{(i,j)} + 1 \{M \text{ is } \mathbb{R}^{N \times (7 \cdot |\Sigma|^2)}\}$ 
  end for
end for
for  $j \leftarrow 1$  to  $7 \cdot |\Sigma|^2$  do
   $G_j \leftarrow$  information gain of column  $j$  of  $M \{G \text{ is } \mathbb{R}^{1 \times (7 \cdot |\Sigma|^2)}\}$ 
end for
for  $j \leftarrow 1$  to  $7 \cdot |\Sigma|^2$  do
  if  $G_j$  is in the top  $k$  of  $G$  then
     $F \leftarrow F \cup \{j\} \{F \text{ instantiated as } \emptyset\}$ 
  end if
end for
for  $j \leftarrow 1$  to  $|F|$  do
   $F^{(\cdot,j)} \leftarrow M^{(\cdot, F_j)} \{F \text{ is } \mathbb{R}^{n \times |F|}\}$ 
end for
return  $F$ 

```

idea for the resulting features is simply to treat the number of occurrences of each 2-let as a feature of the sequence. This results in exactly $7 \cdot \frac{|\Sigma|(|\Sigma|-1)}{2}$ possible features which is a swiftly increasing function of the number of features, i.e., dimensions. Thus, it is necessary to perform feature selection afterwards which is achieved by applying information gain as a feature selection criterion. The algorithm for 2-let feature extraction is summarized in Algorithm 1. To count all 2-let occurrences, an $N \times 7 \cdot |\Sigma|^2$ matrix is used (one row per sequence), which is denoted as SM .

Complexity For each sequence all correctly ordered pairs need to be examined, which amounts to a $\Theta(m^2)$ runtime per e-sequence. Thus, the runtime for shapelet occurrence counting is $\Theta(N \cdot m^2)$, while the memory footprint is $\Theta(N \cdot |\Sigma|^2)$. Calculating information gain of a numeric attribute requires $\Theta(N \cdot \log(N))$ runtime. This is performed for each feature, which means the total runtime of feature selection via information gain is $\Theta(N \cdot \log(N) \cdot |\Sigma|^2)$. Memory remains at $\Theta(N \cdot |\Sigma|^2)$. Thus, putting the two steps together we arrive at $\Theta(N \cdot (m^2 + \log(N) \cdot |\Sigma|^2))$ for runtime and $\Theta(N \cdot |\Sigma|^2)$ memory to execute 2-let extraction and select the best 2-lets as features. Calculating the occurrences of the selected 2-lets for a new e-sequence takes $\Theta(m^2)$ time in the worst case, since once again all its correctly ordered event-interval pairs need to be considered. Note that this is always independent of $|\Sigma|$, since a constant number of 2-lets are selected in the feature selection step.

4.4 Interval segment features (e-lets)

The features we have discussed so far do not capture relations between more than single event-interval pairs. As such, these features are ignoring potentially class-

Fig. 3 The seven temporal relations between an ordered pair of event-intervals (A , B) as defined in Allen's temporal logic

Fig. 4 **a** A follow-by 2-let relation between the events A and B . **b** An e-let capturing the temporal arrangement between event A and B , while also considering the absence of the C event

predictive information from event subsequences that occur concurrently. In addition, class-predictive information related to varied event time spans cannot be fully captured by utilizing the previous three feature types. These limitations are demonstrated in Fig. 4.

To address the aforementioned deficiencies, we introduce the novel concept of e-lets, corresponding to event-interval subsequence-based features, inspired by time series shapelets (Ye and Keogh 2009). We motivate the use of random e-let

Algorithm 2: Extract random e-let-features.

Require: Let $\mathcal{D} = \{S_1, \dots, S_N\}$ be the set of e-sequences, and θ the number of e-let to extract

for $i \leftarrow 1$ **to** θ **do**

$S_j \leftarrow$ random e-sequences from $\mathcal{D}$ with $j \sim \mathcal{U}(0, n)$

$S_j^{k,l} \leftarrow$ random e-let from S_j with $l \sim \mathcal{U}(1, |S|)$ and $k \sim \mathcal{U}(0, |S| - l)$

for $i \leftarrow 1$ **to** N **do**

$M^{(i,\theta)} \leftarrow D(S_j^{k,l}, S_i) \{M \text{ is } \mathbb{R}^{N \times \theta}\}$

end for

end for

return M

features due to the evidence generated in Karlsson et al. (2016) and Wistuba et al. (2015) that such features provide both low cost and state-of-the-art classification performance, while capturing local event duration and multi-dimensional information.

Definition 4 (*e-let*) Given an e-sequence $\mathcal{S} = \{S_1, \dots, S_n\}$, the *e-let* $\mathcal{S}^{k,l}$ of $\mathcal{S}$ is defined as the e-sequence containing all event-intervals that are active from time point k until time point $k + l - 1$. In other words, $\mathcal{S}^{k,l} = \{S_1^*, \dots, S_n^*\}$, where each S_i^* is the time-truncated counterpart of each S_i , such that $S_i^*.t_{ts} = \max\{S_i.t_{ts}, k\}$ and $S_i^*.t_{te} = \min\{S_i.t_{te}, k + l - 1\}$.

The feature extraction process for e-lets proceeds as follows. We first, select θ random e-lets from the training dataset $\mathcal{X}$ uniformly at random. In Algorithm 2, at each iteration i , an e-let $\mathcal{S}_{i_i}^{k_i, l_i}$ is extracted by randomly selecting an e-sequence $S_{i_i} \in \mathcal{X}$, a random starting time point k_i and a random length l_i . Note that the length of e-lets is constrained to be smaller than or equal to the length of the shortest e-sequence in $\mathcal{X}$, i.e., $l_i \leq \min_{S_i \in \mathcal{X}} \{|S_i|\}$. Let us denote this maximum bound on the e-let length as λ . Performing this selection procedure θ times produces a pool of candidate e-lets, which we denote as $\mathcal{E}_{\mathcal{X}, \theta} = \{\mathcal{S}_{i_i}^{k_i, l_i}, i \in [1, \theta]\}$.

Next, we compute the distance of each e-let in $\mathcal{E}_{\mathcal{X}, \theta}$ to each e-sequence in $\mathcal{X}$. Effectively, this produces a mapping of each e-sequence to a set of θ feature values, each corresponding to the distance of the e-sequence to an e-let. As a distance function we employ ABIDE (Kostakis and Gionis 2015), which is an extension of the IBSM distance metric defined for subsequence matching of event-interval sequences. The key structure used by ABIDE is a vector representation of the active event-intervals in an e-sequence at each time point.

Definition 5 (*Active event vector*) Given e-sequence $\mathcal{S}$ and time-point t , the *active event vector* of $\mathcal{S}$ at t , is a $|\Sigma|$ -dimensional binary vector $V_{\mathcal{S}}^t$, such that $V_{\mathcal{S}}^t(i) = 1$, if $e_i \in \Sigma$ is active in $\mathcal{S}$ and 0, otherwise.

Hence, $\mathcal{S}$ can be represented as an ordered set of active event vectors $\mathcal{V}_{\mathcal{S}} = \{V_{\mathcal{S}}^1, \dots, V_{\mathcal{S}}^n\}$. Moreover, the set of distinct event labels that are contained in $\mathcal{S}$, or in other words the set of event labels that are active for at least one time point, is denoted as $\Sigma_{\mathcal{S}}$, where $\Sigma_{\mathcal{S}} \subseteq \Sigma$.

Given an e-sequence $\mathcal{S}$ and a query e-let Q , the ABIDE distance function is then defined simply as the minimum distance of Q to any e-let $\mathcal{S}_{l_i}^{k_i, l_i}$ of $\mathcal{S}$, with $l_i = |Q|$. Computing the ABIDE distance is done by employing a sliding window W_j of length $|Q|$ over $\mathcal{S}$, with j indicating the starting time point of the sliding window. The goal is to find the e-let, that minimizes the distance between the vector-based representation of Q , $\mathcal{V}_Q$, and the L_1 norm of the vector-based representation of the corresponding e-let contained in $\mathcal{V}_{W_j}$. Or more formally:

$$D(Q, \mathcal{S}) = \min_j \sum_{j=1}^{|\mathcal{S}|-|Q|+1} \sum_{t=1}^{|Q|} \sum_{i=1}^{|\Sigma_Q|} |\mathcal{V}_Q^t(i) - \mathcal{V}_{W_j}^t(i)|.$$

The full computation of ABIDE is performed by applying a skyline lower-bounding approach using early-abandoning and alphabet reduction heuristics so as to speed up the computation of the sliding windows vector-based distance. More details about ABIDE can be found in Kostakis and Papapetrou (2017). The resulting set of θ distance values correspond to the features that can then be passed on to the chosen classification model.

Complexity In the worst case, for each generated e-let and each sliding window of length at most λ , ABIDE computes the L_1 norm of Σ -dimensional vectors, for a total of $|\mathcal{S}| - \lambda + 1$ times. This is furthermore performed across all N e-sequence training examples. The total process is executed for each of the total number of generated random shapelets θ , resulting in $\Theta(\theta \cdot N \cdot |\Sigma| \cdot \lambda (|\mathcal{S}| - \lambda + 1)) = \Theta(\theta \cdot N \cdot |\Sigma| \cdot \lambda \cdot |\mathcal{S}|)$ runtime.

5 Experimental evaluation

5.1 Experimental setup

5.1.1 Data

The data consists of information about diagnoses and prescribed drugs for 1,314,646 patients gathered from the research infrastructure Swedish Health Record Research Bank, Health Bank at Stockholm University (Dalianis et al. 2015); an anonymized patient record based on the TakeCare EPR records from Karolinska University Hospital in Stockholm, Sweden. Diagnoses are encoded using the International Statistical Classification of Diseases and Related Health Problems, 10th Edition (ICD-10) and drugs are encoded using the Anatomical Therapeutic Chemical Classification System (ATC). The TakeCare electronic health record data consisted of 84,100,593 entries of ICD10 diagnosis codes, lab test data, and ATC coded medication data.

The classification task chosen is in regard to identifying adverse drug events (ADEs) diagnoses which, according to Nebeker et al. (2004), are injuries that result from the use of a drug, including harm caused by the normal use of a drug, drug overdoses, and use-related harms such as from drug dose reductions and discontinuations of drugs administration. ADEs possess high clinical relevancy being that they account

Table 1 ADE case groups (left) and corresponding control groups (right)

Case groups	Control groups
D611: drug-induced aplastic anaemia	D619: aplastic anemia, unspecified
D642: drug-induced secondary sideroblastic anaemia	D649: anemia, unspecified
D695: secondary thrombocytopenia	D696: thrombocytopenia, unspecified
E273: drug-induced adrenocortical insufficiency	E274: unspecified adrenocortical insufficiency
G620: drug-induced polyneuropathy	G629: polyneuropathy, unspecified
I952: drug-induced hypotension	I959: hypotension, unspecified
L270: drug-induced generalized skin eruption	L309: dermatitis, unspecified
L271: drug-induced localized skin eruption	L309: dermatitis, unspecified
O355: maternal care for (suspected) damage to fetus by drugs	O359: maternal care for (suspected) fetal abnormality and damage, unspecified
R502: drug-induced fever	R509: fever, unspecified
T782: adverse effects: anaphylactic shock	R579: shock, unspecified
T783: adverse effects: angioneurotic edema	R609: edema, unspecified
T801: vascular complications following infusion, transfusion and therapeutic injection	T828: other specified complications of cardiac and vascular prosthetic devices, implants and grafts
T886: drug-induced anaphylactic shock	T780: anaphylactic reaction due to food

for approximately 3.7% of hospital admissions around the world according to Howard et al. (2007).

The diagnosis and prescription data were pre-processed into 14 ADE datasets consisting of interval sequences of ADE case groups (i.e., those patients experiencing an ADE) and control groups (i.e., those patients who do not). The case groups were chosen based on ICD10 codes of known ADEs. Patients diagnosed with these codes were selected as examples consisting of their 90 day diagnosis and medication histories of events occurring before the ADE, from which interval sequences were constructed. For patients whom had more than one occurrence of a particular ADE type, only the last ADE window was included. The specific ADE codes for case and the corresponding selected control group included in the experiments are presented in Table 1.

Corresponding control group codes were chosen based on codes that possessed the greatest medical similarity to the case ADE but did not constitute an actual ADE. Such intervals were given the alternative class labeling and extracted in the same manner as the case groups in which 90 day medical history windows were extracted per example.

The alphabet size of the intervals was reduced by selecting the 200 most frequent ICD10 and ATC codes for each combined case and control group. Candidate ADEs were also excluded from investigation if, after preprocessing, the total number of case and control data intervals was insufficient, in which a threshold of fewer than 2000 intervals was chosen. A basic description of these novel ADE data sets can be observed in Table 2. In addition, experiments were performed on six publicly available single-label benchmark data sets from a variety of domains which include the following: Auslan2 (Mörchen and Fradkin 2010); Blocks (Mörchen and Fradkin

Table 2 Summary of ADE data sets

Dataset	<i>n</i>	# of intervals	# of case	# of control	Majority vote
D611	619	43,056	573	46	0.926
D642	5881	247,268	317	5562	0.946
D695	2361	132,545	1253	1108	0.531
E273	66	2675	27	39	0.591
G620	295	9799	54	241	0.817
I952	402	15,303	50	352	0.876
L270	537	19,834	297	240	0.553
L271	354	12,990	112	242	0.684
O355	247	6032	94	153	0.619
R502	3324	138,819	77	3247	0.977
T782	54	2169	16	38	0.704
T783	216	7523	50	166	0.769
T801	482	21,988	41	441	0.915
T886	59	2366	46	13	0.780

2010); Context (Mörchen and Fradkin 2010); Hepatitis (Patel et al. 2008); Pioneer (Mörchen and Fradkin 2010); and Skating (Mörchen and Fradkin 2010). Multi-labeled datasets examined in Bornemann et al. (2016), which permit a sequence to possess multiple class-labels, were not included in this study.

5.1.2 Parameter configurations

Our framework evaluations have focused on three configuration approaches, all of which were under 10-fold cross validation. The first approach examined which classification model could produce the best performance under SMILE. This model type would then be utilized for the remaining configurations. Our second approach examines the effect on performance of different features added sequentially. Finally, when comparing the predictive performance of the different number of features for STIFE and SMILE, we employ a one-variable-at-a-time design. Specifically, for STIFE we vary one component of the framework, the number of 2-lets, from the set {10, 25, 50, 75, 100, 200}. For SMILE we vary the novel e-let component of the framework, keeping the number of 2-lets constant at 75 features, while varying the number of e-lets from the set {10, 25, 50, 75, 100, 200}. We observe the effects of altering the number of 2-lets and e-lets both on average and across all data sets.

5.1.3 Evaluation metrics

We examined accuracy, i.e., the fraction of correct predictions produced by our classifiers, alongside area under curve (AUC). We focus on AUC as a more meaningful

Table 3 Comparison of classification models for SMILE

Dataset	Accuracy				Area under ROC			
	RF	LR	DT	SVM	RF	LR	DT	SVM
Auslan2	0.485	0.335	0.465	0.305	0.686	0.668	0.674	0.696
Blocks	1.000	0.995	0.995	0.986	1.000	0.994	0.975	1.000
Context	0.988	0.979	0.963	0.996	1.000	0.999	0.969	0.997
Hepatitis	0.831	0.735	0.769	0.823	0.890	0.789	0.788	0.813
Pioneer	0.988	0.981	0.950	0.969	1.000	1.000	0.894	0.969
Skating	0.977	0.970	0.972	0.847	0.999	0.999	0.973	0.966
D611	0.945	0.874	0.921	0.929	0.878	0.744	0.628	0.523
D642	0.988	0.976	0.981	0.976	0.991	0.943	0.897	0.793
D695	0.745	0.726	0.688	0.729	0.813	0.793	0.659	0.726
E273	0.586	0.619	0.581	0.607	0.690	0.628	0.580	0.517
G620	0.854	0.750	0.854	0.816	0.902	0.661	0.767	0.500
I952	0.876	0.751	0.826	0.876	0.670	0.543	0.568	0.500
L270	0.672	0.611	0.605	0.561	0.734	0.637	0.611	0.509
L271	0.746	0.647	0.709	0.692	0.800	0.611	0.624	0.521
O355	0.875	0.713	0.839	0.826	0.944	0.773	0.826	0.853
R502	0.977	0.964	0.975	0.977	0.816	0.591	0.507	0.500
T801	0.917	0.892	0.896	0.913	0.819	0.623	0.716	0.501
T782	0.750	0.763	0.727	0.723	0.508	0.628	0.575	0.500
T783	0.857	0.686	0.732	0.796	0.750	0.568	0.642	0.558
T886	0.771	0.788	0.838	0.786	0.834	0.759	0.696	0.502
Avg.	0.841	0.788	0.814	0.807	0.836	0.748	0.728	0.672

Bold represents the best performing method

performance measure throughout our evaluation due to the considerable class imbalance which can be observed in many of our chosen data sets, as seen in Table 2.

5.2 Empirical investigation

5.2.1 Model comparison

We chose to begin our investigation by examining which classification model type would yield the best performance outcome while incorporating all stages, i.e. up to Phase SMILE, of our novel framework. For model comparisons, Friedman tests showed a highly significant result, for both accuracy ($\chi^2_F = 19.545$, $df = 3$, $p = 0.0002109$) and for AUC ($\chi^2_F = 31.155$, $df = 3$, $p = 0.0000007885$) across Random Forest (RF), Logistic Regression (LR), Decision Tree (DT), and Support Vector Machine (SVM) algorithms which incorporate all four stages of SMILE. A full comparison of model types across all data sets can be viewed in Table 3.

Shown in Fig. 5 are critical distance plots for the post-hoc Nemenyi tests with $\alpha = 0.05$. Such plots were first used for visualization purposes by Demšar (2006). We

Fig. 5 Critical distance plots of Nemenyi test model comparison for $\alpha = 0.05$, shown both for Accuracy (left figure), and AUC (right figure). Groups of classifiers which are not significantly different are shown as connected

Table 4 Wilcoxon signed-rank test between STIFE and SMILE

Methods	Accuracy		Area under ROC	
	z-score	p value	z-score	p value
STIFE versus SMILE	-1.29	0.201	-2.15	0.032 [†]

can observe that the RF model is significantly better than all other models in regard to critical distance determined for both accuracy and AUC. For both accuracy and AUC there was no significant difference when comparing LR, DT, and SVM models. Based on this evidence, RF was selected as the best performing model, and thus, it remains the chosen model for all further analyses.

5.2.2 Comparison of methods

Shown in Table 4 are the paired-difference Wilcoxon signed-rank test results comparing SMILE to STIFE. We observe that SMILE yielded significantly ($p < 0.05$) different population mean rank results over STIFE in terms of AUC, however improvements were insignificant when examining accuracy. Although improvements are not as pronounced in terms of accuracy, we once again emphasize the importance of AUC as the most valid metric in this study due to the profound class imbalances of our data sets. As such, this finding strongly indicates the superior approach of e-let feature inclusion, due to the addition of class-predictive information stemming from the use of e-lets.

Secondly, we employed Friedman tests to allow for multiple comparisons between all competitor methods. In addition to utilizing the various stages of STIFE as competitors, we also utilize 1-nearest neighbor under IB SM distance, alongside using SPAM as an appropriate competitor from the sequential pattern mining domain. SPAM was initialized with a minsup of 0.1, minimum pattern length of 2, minimum pattern length of 8, and a max gap of 2. For method comparisons, Friedman tests showed highly significantly different results ($p < 0.01$) for both accuracy ($\chi^2_F = 52.964$, $df = 5$, $p = 0.000000003421$) and for AUC ($\chi^2_F = 35.45$, $df = 4$, $p = 0.0000003755$)

Fig. 6 Critical distance plots of Nemenyi test method comparison for $\alpha = 0.05$, shown for Accuracy (left), and $\alpha = 0.01$ shown for AUC (right). Groups of methods which are not significantly different are shown as connected

Table 5 Area under ROC comparison across all methods

Dataset	AUC				
	SPAM	STATIC	MEDOID	STIFE	SMILE
Auslan2	0.769	0.585	0.748	0.631	0.686
Blocks	0.510	0.899	0.900	1.000	1.000
Context	0.968	0.994	1.000	1.000	1.000
Hepatitis	0.757	0.695	0.836	0.862	0.890
Pioneer	0.734	0.892	0.964	1.000	1.000
Skating	0.955	0.914	0.977	0.997	0.999
D611	0.740	0.730	0.711	0.753	0.878
D642	0.960	0.926	0.927	0.989	0.991
D695	0.706	0.704	0.704	0.809	0.813
E273	0.640	0.636	0.617	0.597	0.690
G620	0.756	0.702	0.682	0.902	0.902
I952	0.555	0.550	0.550	0.584	0.642
L270	0.542	0.528	0.541	0.686	0.734
L271	0.683	0.650	0.634	0.786	0.800
O355	0.852	0.833	0.826	0.948	0.944
R502	0.693	0.752	0.739	0.771	0.816
T782	0.590	0.702	0.727	0.717	0.508
T783	0.528	0.752	0.752	0.777	0.750
T801	0.701	0.738	0.804	0.804	0.819
T886	0.831	0.692	0.667	0.711	0.834
Avg.	0.724	0.744	0.765	0.816	0.835

Bold represents the best performing method

Table 6 Accuracy comparison across all methods

Dataset	Accuracy					
	SPAM	STATIC	MEDOID	IBSM	STIFE	SMILE
Auslan2	0.320	0.305	0.460	0.400	0.505	0.485
Blocks	0.138	0.933	1.000	1.000	1.000	1.000
Context	0.888	0.954	0.983	0.971	0.992	0.988
Hepatitis	0.697	0.663	0.765	0.765	0.803	0.831
Pioneer	0.773	0.813	0.900	0.956	0.981	0.988
Skating	0.813	0.674	0.919	0.983	0.966	0.977
D611	0.940	0.937	0.940	0.813	0.948	0.945
D642	0.972	0.959	0.960	0.948	0.988	0.988
D695	0.670	0.644	0.644	0.556	0.744	0.745
E273	0.621	0.631	0.602	0.324	0.569	0.586
G620	0.786	0.831	0.807	0.681	0.858	0.854
I952	0.858	0.871	0.871	0.779	0.866	0.876
L270	0.542	0.515	0.525	0.525	0.637	0.672
L271	0.732	0.687	0.684	0.647	0.758	0.746
O355	0.745	0.770	0.766	0.713	0.867	0.875
R502	0.975	0.976	0.977	0.962	0.976	0.977
T782	0.648	0.683	0.667	0.243	0.723	0.750
T783	0.740	0.778	0.760	0.710	0.834	0.857
T801	0.894	0.909	0.907	0.880	0.915	0.917
T886	0.778	0.733	0.750	0.671	0.900	0.871
Avg.	0.727	0.763	0.794	0.730	0.842	0.846

Bold represents the best performing method

among competitors. Shown in Fig. 6 are critical distance plots for the post-hoc Nemenyi tests with $\alpha = 0.01$ for AUC (right), demonstrating on a highly significant level that SMILE outperforms the MEDOID, SPAM, and STATIC methods while STIFE does not outperform either SPAM nor MEDOID on a highly significant level. For accuracy (left), we provide evidence given $\alpha = 0.05$, that SMILE outperforms the SPAM, 1-nearest neighbor using IBSM, MEDOID, and STATIC methods while STIFE is unable to outperform the MEDOID method.

Examining average AUC and accuracy performance across all datasets in Tables 5 and 6 also highlights the superior performance of SMILE.

5.2.3 Effect of number of 2-lets and e-lets

Figure 7 demonstrates average performance metrics comparison across all data sets regarding 2-let features with values from the set $\{10, 25, 50, 75, 100, 200\}$ for STIFE. Also seen are the e-let feature variations chosen from the same set of values. For SMILE, the number of 2-lets was kept constant at 75. The effect of varied numbers of features demonstrated relatively little variation in performance,

Fig. 7 Average performance metrics comparison across all data sets of STIFE shown for variable number of 2-let and e-let features across values {10, 25, 50, 75, 100, 200} for AUC (left) and Accuracy (right). For SMILE the number of 2-lets was kept constant at 75

regardless of the configuration examined. Examining AUC for STIFE, a trend can be observed for small performance improvements from a greater number of features, while for SMILE the best AUC was reported for 100 e-let features.

In Figs. 8, 9, 10, and 11 we similarly examine the effect of the number of features on performance for each dataset available and observe a greater variation in results when examining the novel ADE data sets, where selecting either the lowest or highest number of features could result in the best performance depending on the dataset. Based on this evidence we regard the use of 75 2-let and e-let features to be reasonable, and we would motivate future examinations to choose a similarly conservative number of both features to reduce cost.

5.3 Medical case study on e-let features

To further motivate the utility of including e-let features in SMILE, we examine several e-lets which were ranked in the top 10 of feature importance for a given data set i.e., contributing the highest average impurity decrease for the RF classifier across all feature types. Figures 12, 13, and 14 depict examples of real e-lets extracted from patients which SMILE prioritizes as being highly ranked in regard to feature importance. Due to the inherent randomness of these extracted e-lets, we do not suggest that each interval contained in the e-lets is of importance to class discrimination. With this in mind, an e-let with a subset of intervals in the provided examples could possess equal feature importance. Seen in Figs. 12 and 14 are two e-lets of highly ranked importance generated from the ADED611 medical data set, which discriminate between drug-induced aplastic anemia and unspecified aplastic anemia. Examining Fig. 12 reveals that the drugs Omeprazole and Furosemide prescribed over an extended duration contribute to high importance. This finding is backed up by medical literature which reports that Furosemide is a diuretic drug prescribed to cardiovascular patients and has a known association to drug-induced aplastic anemia (Rao 2014). Omeprazole

Fig. 8 Performance metrics comparison of STIFE and SMILE shown for variable number of 2-let and e-let features across all ADE datasets. For SMILE the number of 2-lets was kept constant at 75

Fig. 9 Performance metrics comparison of STIFE and SMILE shown for variable number of 2-let and e-let features across across all benchmark datasets. For SMILE the number of 2-lets was kept constant at 75

is a proton-pump inhibitor with one study reporting a link of such inhibitors to anemia in cardiovascular outpatients (Shikata et al. 2014). Although Omeprazole has not been proven to be linked to drug-induced aplastic anemia in particular, our finding suggests it might contribute to the condition if its prescription occurs alongside Furosemide.

Secondly we examine Fig. 13 showing an e-let of high importance, extracted from a patient with long term multiple myeloma, whom also had treatments of anti-neoplastic chemotherapy and immunotherapy for this cancer. This finding might suggest that drug-induced aplastic anemia is more likely for this particular cancer combined with the respective chemotherapy and immunotherapy regimens for treatment. Finally, we examine Fig. 14 showing an e-let of high importance extracted from a patient initially possessing the prescription of 6 drugs and was later diagnosed with lymphocytic leukemia. Of the drugs under examination, the antibiotic trimethoprim-sulfamethoxazole has been known to induce aplastic anemia (Menger et al. 2015) while lansoprazole, a proton pump inhibitor, has been linked to drug-induced hemolytic anemia (Rao 2014). In this example, although there is greater ambiguity regarding how the contribution of drugs and leukemia diagnosis provides high importance, such a finding may be of medical interest.

Fig. 10 Performance metrics comparison of STIFE and SMILE shown for variable number of 2-let and e-let features across across all ADE datasets. Baseline refers to the majority vote baseline shown in Table 2. For SMILE the number of 2-lets was kept constant at 75

Fig. 11 Performance metrics comparison of STIFE and SMILE shown for variable number of 2-let and e-let features across across all benchmark datasets. For SMILE the number of 2-lets was kept constant at 75

Fig. 12 Example of e-let extracted for the ADED611 medical data set suggesting high importance relating to Omeprazole and Furosemide drug prescriptions over an extended duration

Fig. 13 Example of e-let extracted for the ADED611 medical data set suggesting high importance for multiple myeloma over a long duration containing antineoplastic chemotherapy and immunotherapy over a shorter duration

Fig. 14 Example of e-let extracted for the ADED611 medical data set suggesting a combination of the many observed drugs being of high importance

6 Conclusions

The main contribution of this paper is the introduction of the SMILE framework, which was motivated by a need to capture information regarding event duration across multiple event types within e-sequences. A comprehensive evaluation has been performed which demonstrates that SMILE provides significantly improved AUC performance over the current state-of-the-art, alongside a selection of competitors utilising varied combinations of features types contained within the SMILE framework itself. This evaluation was performed across a series of benchmark and newly generated ADE data sets. The investigation also reveals that the selection of the random forest model for use with SMILE achieved significantly better performance over a variety of competitor classifiers. Finally, the investigation demonstrated the effect of utilising a varied number of SMILE features, with the result being that a conservative number of features was often appropriate to achieve the best results. Such findings contribute to a growing knowledge base of informative features which can be employed for sequences of temporal intervals to achieve state-of-the-art performance for a variety of domains such as ADE detection. Directions for future work involve: investigating approaches to reduce feature extraction costs, utilising alternative similarity measures, more extensive medical validation of important features, examining the applicability of our framework in alternative domains, and introducing methods to aid in the interpretability of our framework.

Funding Open access funding provided by Stockholm University.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Funding This work was partly supported by the VR-2016-03372 Swedish Research Council Starting Grant as well as by grants provided by Stockholm University and Stockholm County Council (SU-SLL). Ethical approval was granted by the Stockholm Regional Ethical Review Board under Permission No. 2012/834-31/5.

Source code The source code for the implementation of SMILE can be found at: <https://gitlab.com/isakkarlsson/smile/>.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Allen JF (1983) Maintaining knowledge about temporal intervals. *Commun ACM* 26(11):832–843
- Ayres J, Flannick J, Gehrke J, Yiu T (2002) Sequential pattern mining using a bitmap representation. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp 429–435
- Batal I, Valizadegan H, Cooper GF, Hauskrecht M (2013) A temporal pattern mining approach for classifying electronic health record data. *ACM Trans Intell Syst Technol (TIST)* 4(4):63
- Bornemann L, Lecerf J, Papapetrou P (2016) Stife: a framework for feature-based classification of sequences of temporal intervals. In: *International conference on discovery science*. Springer, pp 85–100
- Dalianis H, Henriksson A, Kvist M, Velupillai S, Weegar R (2015) Health bank: a workbench for data science applications in healthcare. In: *CAiSE-2015 industry track co-located with 27th conference on advanced information systems engineering (CAiSE-CEUR)*, vol 1381, pp 1–18
- Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7(Jan):1–30
- Giannotti F, Nanni M, Pedreschi D (2006) Efficient mining of temporally annotated sequences. In: *Proceedings of the 6th SIAM data mining conference*, vol 124, pp 348–359
- Höppner F, Klawonn F (2001) Finding informative rules in interval sequences. In: *Proceedings of the 4th international symposium on advances in intelligent data analysis*, pp 123–132
- Howard R, Avery A, Slavenburg S, Royal S, Pipe G, Lucassen P, Pirmohamed M (2007) Which drugs cause preventable admissions to hospital? A systematic review. *Br J Clin Pharmacol* 63(2):136–147
- Karlsson I, Boström H (2016) Predicting adverse drug events using heterogeneous event sequences. In: *2016 IEEE international conference on healthcare informatics (ICHI)*. IEEE, pp 356–362
- Karlsson I, Papapetrou P, Boström H (2016) Generalized random shapelet forests. *Data Min Knowl Discov* 30(5):1053–1085. <https://doi.org/10.1007/s10618-016-0473-y>
- Kosara R, Miksch S (2001) Visualizing complex notions of time. *Stud Health Technol Inform* 1:211–215
- Kostakis O, Papapetrou P (2017) Abide: querying time-evolving sequences of temporal intervals. In: *International symposium on intelligent data analysis*. Springer, pp 173–185
- Kostakis O, Papapetrou P, Hollmén J (2011) Artemis: assessing the similarity of event-interval sequences. In: *Proceedings of the conference on machine learning and knowledge discovery in databases (ECML/PKDD 2011)*, pp 229–244
- Kostakis OK, Gionis AG (2015) Subsequence search in event-interval sequences. In: *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. ACM, pp 851–854
- Kotsifakos A, Papapetrou P, Athitsos V (2013) Ibsm: interval-based sequence matching. In: *Proceedings of the SIAM conference on data mining (SDM)*
- Lam HT, Mörchén F, Fradkin D, Calders T (2014) Mining compressing sequential patterns. *Stat Anal Data Min ASA Data Sci J* 7(1):34–52
- Laxman S, Sastry P, Unnikrishnan K (2007) Discovering frequent generalized episodes when events persist for different durations. *IEEE Trans Knowl Data Eng* 19(9):1188–1201. <https://doi.org/10.1109/TKDE.2007.1055>
- Lin JL (2003) Mining maximal frequent intervals. In: *Proceedings of the 18th ACM symposium on applied computing*, pp 624–629

- Menger RP, Dossani RH, Thakur JD, Farokhi F, Morrow K, Guthikonda B (2015) Extra-axial hematoma and trimethoprim-sulfamethoxazole induced aplastic anemia: the role of hematological diseases in subdural and epidural hemorrhage. *Case reports in hematology* 2015
- Mooney C, Roddick JF (2004) Mining relationships between interacting episodes. In: *Proceedings of the 4th SIAM international conference on data mining*
- Mörchen F, Fradkin D (2010) Robust mining of time intervals with semi-interval partial order patterns. In: *Proceedings of the 2010 SIAM international conference on data mining, SIAM*, pp 315–326
- Moskovitch R, Shahar Y (2015a) Classification-driven temporal discretization of multivariate time series. *Data Min Knowl Discov* 29(4):871–913
- Moskovitch R, Shahar Y (2015b) Classification of multivariate time series via temporal abstraction and time intervals mining. *Knowl Inf Syst* 45(1):35–74
- Moskovitch R, Walsh C, Wang F, Hripcsak G, Tatonetti N (2015) Outcomes prediction via time intervals related patterns. In: *2015 IEEE international conference on data mining (ICDM)*. IEEE, pp 919–924
- Nebeker JR, Barach P, Samore MH (2004) Clarifying adverse drug events: a clinician's guide to terminology, documentation, and reporting. *Ann Intern Med* 140(10):795–801
- Pachet F, Ramalho G, Carrive J (1996) Representing temporal musical objects and reasoning in the MusES system. *J New Music Res* 25(3):252–275
- Papapetrou P, Kollios G, Sclaroff S, Gunopulos D (2005) Discovering frequent arrangements of temporal intervals. In: *Proceedings of 5th IEEE international conference on data mining*, pp 354–361
- Papapetrou P, Kollios G, Sclaroff S, Gunopulos D (2009) Mining frequent arrangements of temporal intervals. *Knowl Inf Syst* 21:133–171
- Patel D, Hsu W, Lee ML (2008) Mining relationships among interval-based events for classification. In: *Proceedings of the 2008 ACM SIGMOD international conference on management of data*. ACM, pp 393–404
- Rao KV (2014) eChapter 24. Drug-induced hematologic disorders. The McGraw-Hill Companies, New York. <https://accesspharmacy.mhmedical.com/content.aspx?aid=57525021>
- Shikata T, Sasaki N, Ueda M, Kimura T, Itohara K, Sugahara M, Fukui M, Manabe E, Masuyama T, Tsujino T (2014) Use of proton pump inhibitors is associated with anemia in cardiovascular outpatients. *Circ J* 79(1):193–200. <https://doi.org/10.1253/circj.CJ-14-0582>
- Uddin MT, Uddiny MA (2015) Human activity recognition from wearable sensors using extremely randomized trees. In: *2015 International conference on electrical engineering and information communication technology (ICEEICT)*, pp 1–6
- Winarko E, Roddick JF (2007) Armada: an algorithm for discovering richer relative temporal association rules from interval-based data. *Data Knowl Eng* 63(1):76–90. <https://doi.org/10.1016/j.datak.2006.10.009>
- Wistuba M, Grabocka J, Schmidt-Thieme L (2015) Ultra-fast shapelets for time series classification. *arXiv preprint arXiv:1503.05018*
- Yan X, Han J, Afshar R (2003) Clospan: mining: closed sequential patterns in large datasets. <https://doi.org/10.1137/1.9781611972733.15>
- Ye L, Keogh E (2009) Time series shapelets: a new primitive for data mining. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp 947–956
- Zhou C, Cule B, Goethals B (2015) Pattern based sequence classification. *IEEE Trans Knowl Data Eng* 28(5):1285–1298