Proceedings of the Federated Conference on Computer Science and Information Systems pp. 47–56
DOI: 10.15439/2018F19
ISSN 2300-5963 ACSIS, Vol. 15

Efficient Discovery of Top-K Sequential Patterns in Event-Based Spatio-Temporal Data

Piotr S. Maciąg
Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19,
00-665, Warsaw, Poland,

***Abstract*—We consider the problem of discovering sequential patterns from event-based spatio-temporal data. The dataset is described by a set of event types and their instances. Based on the given dataset, the task is to discover all significant sequential patterns denoting the attraction relation between event types occurring in a pattern. Already proposed algorithms discover all significant sequential patterns based on the significance threshold, which minimal value is given by an expert. Due to the nature of described data and complexity of discovered patterns, it may be very difficult to provide reasonable value of significance threshold. We consider the problem of effective discovering K most important patterns in a given dataset (that is, discovering top-K patterns). We propose algorithms for unlimited memory environments. Developed algorithms have been verified using synthetic and real datasets.**

I. Introduction

DISCOVERING knowledge from spatio-temporal data is gaining attention of researchers nowadays. Based on literature, we can distinguish two basic types of spatio-temporal data: event-based and trajectory-based [1]. Event-based spatio-temporal data is described by a set of event types $F = \{f_1, f_2, \ldots, f_n\}$ and a set of instances D. Each instance $e \in D$ denotes an occurrence of a particular event type from F and is associated with instance identifier, location in spatial dimension and occurrence time. Fig. 1 provides possible sets $D = \{a1, a2, \ldots, d10\}$ and $F = \{A, B, C, D\}$. The same datasets are presented in Table I. Event-based spatio-temporal data and the problem of discovering frequent sequential patterns in this type of data have been introduced in [2].

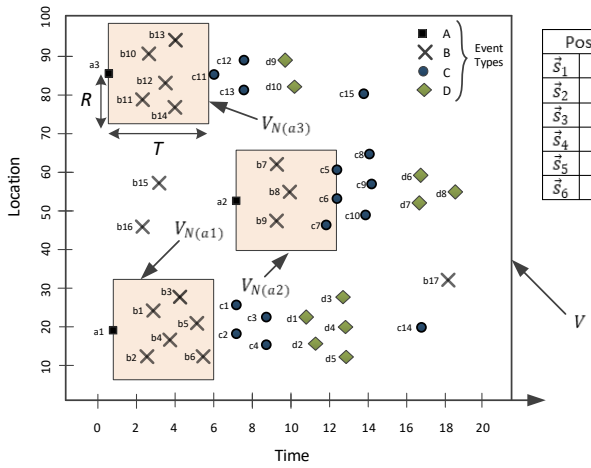
The task of mining spatio-temporal sequential patterns in given datasets F and D may be defined as follows. We assume that the *following* relation (or attraction relation) $f_{i_1} \rightarrow f_{i_2}$ between any two event types $f_{i_1}, f_{i_2} \in F$ denotes the fact, that instances of event type f_{i_1} attract in their spatial and temporal neighborhoods occurrences of instances of event type f_{i_2}. The strength of the following relation $f_{i_1} \rightarrow f_{i_2}$ is investigated by dividing the density of instances of type f_{i_2} in spatio-temporal neighborhoods of instances of type f_{i_1} and density of instances of type f_{i_2} in the whole spatio-temporal embedding space V. If obtained ratio is greater than 1, then it is possible that $f_{i_1} \rightarrow f_{i_2}$ constitute a pattern. We provide the strict definition of density in Section III. The problem introduced in [2] is to discover all significant sequential patterns defined in the form $f_{i_1} \rightarrow f_{i_2} \rightarrow \cdots \rightarrow f_{i_m}$, where the significance threshold is given by an expert. In contrary to this approach, we consider the problem of discovering K most significant patterns in the given dataset. Providing significance threshold for discovering patterns may be difficult due to the complex nature of considered task.

TABLE I
An example of a spatio-temporal event-based dataset

Identifier	Event type	Spatial location	Occurrence time
a1	A	19	1
a2	A	83	1
⋮	⋮	⋮	⋮
b1	B	25	3
b2	B	1	3
⋮	⋮	⋮	⋮
c1	C	25	7
c2	C	15	7
⋮	⋮	⋮	⋮
d1	D	21	11
d2	D	13	12
⋮	⋮	⋮	⋮

The rest of the paper is organized as follows. Related work is described in Section II. In Section III, we provide elementary notions. Our algorithms and main results are presented in Section IV. In Section V, we provide experimental results for both real and synthetic data. In Section VI, we give conclusions and future problems. The main results of the paper are:

1) We introduce the notion of top-K patterns in event-based spatio-temporal data, namely we define the ranking of top-K sequential patterns with minimal length given by parameter *min_len* and point out the efficient pruning strategy for creating the top sequences set.
2) We formulate the algorithm discovering such top sequential patterns in event-based spatio-temporal data.
3) Proposed algorithm has been verified using both synthetic and real datasets. For experiments on synthetic data we used the same types of datasets as used in [2]. As a real datasets, we used the two types of datasets containing event instances related to air pollution data.



Possible significant sequences	
$\vec{s}_1$	A → B → C →D
$\vec{s}_2$	A → B →C
$\vec{s}_3$	B → C →D
$\vec{s}_4$	A → B
$\vec{s}_5$	B → C
$\vec{s}_6$	C → D

Fig. 1. Visualization the spatio-temporal event-based dataset from Table I and a set of possible significant sequences

II. Related Work

The problem of discovering top-K most important frequent patterns in various types of data has been well investigated in literature. TFP algorithm for discovering top-K closed frequent patterns for transaction databases has been given in [3]. In this approach, the user has a possibility to provide a parameter min_len specifying minimal length of discovered patterns (that is, minimal number of items occurring in a pattern). TFP discovers top-K closed frequent patterns by means of the FP-growth algorithm (proposed in [4]) for frequent patterns mining. The authors of [5] extends approach proposed in [3], by considering the problem of effective discovering top-K closed sequential patterns for transaction databases (the notion of closed sequential patterns has been introduced in [6]) and giving algorithm TSP for that purpose. In [7], the authors provide an algorithm discovering top-K jumping emerging patterns.

The problem of discovering sequential patterns in databases containing transactions records has been well investigated. The reader may refer to [8], [9], [10], [3] for the fundamental notions in this topic. More recently, surveys on methods for mining sequential patterns are given in [11], [12]. More recent papers in the area of mining top important frequent itemsets are [13], [14], [15].

Various types of methods have been developed for discovering patterns in event-based spatio-temporal data. The authors of [2] introduce the notion of sequential pattern for event-based spatio-temporal data and provide algorithms for both limited and unlimited memory environments. Obtained results show usefulness of proposed approach, however experiments (i.e. computation time) obtained for large datasets seem to be unsatisfactory. On the other hand, results presented in [2] are not well verified using real datasets. The additional drawback of algorithms proposed in [2] is large number of noise and redundant patterns obtained during mining process. The method of discovering top-K introduced in our article eliminate these deficiencies. A survey of methods for discovering patterns in spatio-temporal data is given in [16], [17]. The problem of discovering hierarchical spatio-temporal patterns has been considered in [18]. The problem of discovering spatio-temporal patterns from trajectory data and objects movements data has been considered in [19], [20], [1], [21], [22], [23].

III. Basic Notions

The dataset given in Fig. 1 is contained in the spatio-temporal space V, which temporal dimension is of size 20 and spatial location is provided by numbers between 0 and 100. For simplicity in Fig. 1 we denote spatial location in only one dimension. Usually, spatial location is defined by two dimensions (f.e. geographical coordinates). By $|V|$ we denote the volume of space V, calculated as the product of spatial area and size of time dimension. Spatial and temporal sizes of spatio-temporal space are usually given by an expert. For example, for Fig. 1 $|V| = 20 * 100 = 2000$. In the following definitions and notions we use terms sequential patterns and sequence interchangeably.

Definition 1. Neighborhood space. By $V_{N(e)}$ we denote the neighborhood space of instance e. For $V_{N(e)}$ having cylindrical shape, R denotes the spatial radius and T temporal interval of that space. The volume $|V_{N(e)}|$ of neighborhood space is equal to $\pi * R^2 * T$.

The shape of $V_{N(e)}$ is given by an expert and may be adjusted to particular dataset. Consider example given in Fig. 1 where we denote neighborhood spaces $V_{N(a1)}, V_{N(a2)}, V_{N(a3)}$. In Fig. 2, we provide an example of cylindrical neighborhood space $V_{N(a1)}$ with spatial location specified by two coordinates. The volume of that space is $|V_{N(a1)}| \approx 384.65$.

The reader may refer to [2] for other possible definitions of neighborhood spaces.

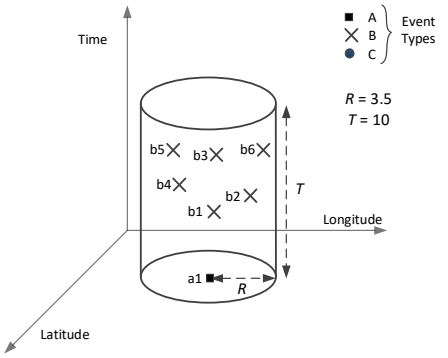


Fig. 2. Possible shape of neighborhood space $V_{N(a1)}$

Definition 2. Neighborhood [2]. For a given event type f and an occurrence of event instance e of that type, the neighborhood of e is defined as follows:

$$\begin{aligned} N(e) = &\{p | p \in D \\ &\wedge distance(p.location, e.location) \leq R \\ &\wedge (p.time - e.time) \in [0, T]\} \end{aligned} \quad (1)$$

where R denotes the spatial radius and T temporal interval of the neighborhood space $V_{N(e)}$.

As the neighborhood $N(e)$ of instance e, we denote the set of instances contained inside the neighborhood space $V_{N(e)}$.The neighborhood of instance $a1$ (shown in Fig. 2) with respect to event type B is $N(a1) = \{b1, b2, b3, b4, b5, b6\}$.

Definition 3. Density [2]. For a given spatiotemporal space V, event type f and its events instances in D, density is defined as follows:

$$Density(f, V) = \frac{|\{e | e.type = f \wedge \text{e is inside V}\}|}{|V|} \quad (2)$$

that is, density is the number of instances of type f occurring inside space V divided by the volume of that space.

Definition 4. Density ratio [2]. Density ratio for two event types f_{i_1}, f_{i_2} and their instances in D is defined as follows:

$$DR(f_{i_1} \rightarrow f_{i_2}) = \frac{avg_{e.type=f_{i_1}}(Density(f_{i_2}, V_{N(e)}))}{Density(f_{i_2}, V)} \quad (3)$$

where $\rightarrow$ denotes the *following* relation between event types f_{i_1}, f_{i_2}.
$avg_{e \in f_{i_1}}(Density(f_{i_2}, V_{N(e)}))$ specifies the average density of instances of type f_{i_2} occurring inside the neighborhood spaces $V_{N(e)}$ created for instances $e \in f_{i_1}$. V denotes the whole considered spatio-temporal space and $Density(f_{i_2}, V)$ specifies density of instances of type f_{i_2} inside that space.

If the value of density ratio for event types f_{i_1} and f_{i_2} is greater than one, then instances of type f_{i_1} attract in their spatio-temporal neighborhood spaces occurrences of instances of type f_{i_2}. If the value is below one, then they repel occurrences of instances of type f_{i_2}. If the value is equal to one, then there is no correlation between these two event types.

Definition 5. Sequence (sequential pattern) $\overrightarrow{s}$ and tailEventSet($\overrightarrow{s}$) [2]. $\overrightarrow{s}$ denotes a m-length sequence of event types: $s[1] \rightarrow s[2] \rightarrow \cdots \rightarrow s[m-1] \rightarrow s[m]$. tailEventSet($\overrightarrow{s}$) denotes the set of instances of type $\overrightarrow{s}[m]$ participating in the sequence $\overrightarrow{s}$.

Consider sequence $\overrightarrow{s_4} = A \rightarrow B$ given in Fig. 1. The length of the sequence is 2 and tailEventSet($\overrightarrow{s_4}$) = $\{b1, b2, \ldots, b14\}$ contains instances of event type B, which are in neighborhoods of instances of event type A.

Definition 6. Sequence index [2]. For a given m-length sequence $\overrightarrow{s}$, sequence index is defined as follows:

1) When $m = 2$ then:

$$SI(\overrightarrow{s}) = DR(\overrightarrow{s}[1] \rightarrow \overrightarrow{s}[2]) \quad (4)$$

2) When $m > 2$ then:

$$SI(\overrightarrow{s}) = \min \begin{cases} SI(\overrightarrow{s}[1:m-1]), \\ DR(\overrightarrow{s}[m-1] \rightarrow \overrightarrow{s}[m]) \end{cases} \quad (5)$$

where sequence $\overrightarrow{s}$ is constituted of event types $\overrightarrow{s}[1] \rightarrow \overrightarrow{s}[2] \rightarrow \cdots \rightarrow \overrightarrow{s}[m]$.

Example 1. Consider the dataset given in Fig .1. As an example let us consider the process of expanding sequence $\overrightarrow{s_1}$. One may notice that density of instances of type B is significant in the neighborhood spaces created for instances of type A. 1-length sequence $\overrightarrow{s_1} = A$ will be expanded to $\overrightarrow{s_1} = A \rightarrow B$ and as the tail event set of $\overrightarrow{s_1}$, the set of instances of type B contained in $N(a1)$ or $N(a2)$ or $N(a3)$ will be remembered (that is, tailEventSet($\overrightarrow{s_1}$) = $\{b1, b2, \ldots, b14\}$). The neighborhood spaces will be created for each instance contained in tailEventSet($\overrightarrow{s_1}$) and $\overrightarrow{s_1}$ will be expanded with event type C, to create $\overrightarrow{s_1} = A \rightarrow B \rightarrow C$. Actual will be tailEventSet($\overrightarrow{s_1}$) = $\{c1, c2, \ldots, c13\}$. In the same manner, the sequence will be expanded with event type D.

The sketch of the ST-Miner algorithm provided in [2] is as follows. First, for each event type in a dataset F, a 1-length sequence is created. Then, in a depth-first manner, each sequence is expanded with each event type in F, if the value of density ratio between the last event type in the sequence and event type considered to be appended is greater than the predefined threshold. The value of density ratio between these two event types is calculated by taking all instances from the tail event set of the sequence, creating their neighborhood spaces and verifying the ratio of the average density of instances of event type considered to be appended in these neighborhood spaces and the total density of instances of that type in the embedding space. If the value of density ratio is below given threshold, then the sequence is not expanded

any more. If the opposite is true, the sequence is expanded in the recursive way. The minimal value of density ratio between any two consecutive event types participating in the sequence is the sequence index (SI($\overrightarrow{s}$)).

IV. Discovering Top-K Patterns

In this section, we provide our algorithms discovering top-K sequential patterns.

Definition 7. For a sequence $\overrightarrow{s} \rightarrow f$ of length $m+1$, we say that f follows event type $\overrightarrow{s}[m]$. tailEventSet($\overrightarrow{s} \rightarrow f$) contains all instances of type f contained in the neighborhoods created for instances from tailEventSet($\overrightarrow{s}$).

Definition 8. Supersequence and subsequence. For two sequences $\overrightarrow{s_i} = \overrightarrow{s_i}[1] \rightarrow \overrightarrow{s_i}[2] \rightarrow \cdots \rightarrow \overrightarrow{s_i}[m_i]$ and $\overrightarrow{s_j} = \overrightarrow{s_j}[1] \rightarrow \overrightarrow{s_j}[2] \rightarrow \cdots \rightarrow \overrightarrow{s_j}[m_j]$, where $m_j > m_i$, $\overrightarrow{s_j}$ is supersequence of $\overrightarrow{s_i}$ ($\overrightarrow{s_i}$ is subsequence of $\overrightarrow{s_j}$) if only $\overrightarrow{s_i}[1] = \overrightarrow{s_j}[1] \wedge \overrightarrow{s_i}[2] = \overrightarrow{s_j}[2] \wedge \cdots \wedge \overrightarrow{s_i}[m_i] = \overrightarrow{s_j}[m_i]$.

In Fig. 1, $\overrightarrow{s_1}$ is supersequence of $\overrightarrow{s_2}$ ($\overrightarrow{s_2}$ is subsequence of $\overrightarrow{s_1}$). Please note however, that for example $\overrightarrow{s_1}$ is not supersequence of $\overrightarrow{s_3}$ (and $\overrightarrow{s_3}$ is not subsequence of $\overrightarrow{s_1}$).

Definition 9. Top-K sequence (sequential pattern). We say that sequence $\overrightarrow{s}$ of length *min_len* is the K-th top sequence (sequential pattern), if there exist K-1 sequences in the top sequences set with length *min_len* and the sequence index of each is equal or greater than $SI(\overrightarrow{s})$.

Definition 10. Pruning threshold θ. Actual pruning threshold θ for sequences considered to be in the top sequences set is equal to the sequence index of any K-th top already discovered sequence.

Lemma 1. For a given sequence $\overrightarrow{s}$ of a minimal length *min_len*, if the sequence index $SI(\overrightarrow{s})$ is below the actual pruning threshold θ, then $\overrightarrow{s}$ and any of its supersequences do not belong to top sequences and $\overrightarrow{s}$ should not be expanded with new event types any more.

Proof: If the sequence index of considered sequence $\overrightarrow{s}$ is below pruning threshold θ, then $\overrightarrow{s}$ does not belong to already discovered top sequences. By means of Definition 6 and Definition 8 any supersequence of $\overrightarrow{s}$ also does not belong to top-K sequences set, so $\overrightarrow{s}$ should not be expanded with new event types. ■

Informally the approach discovering top-K sequences is as follows: starting with 1-length sequences (that is, sequences containing singular event types) expand each sequence in a depth-first manner up to the moment when its length is at least *min_len*. We start discovering sequences with the basic value of pruning threshold θ equal to 1. At the same time we maintain the set of top-K already discovered patterns. By $D(f)$ we denote set of instances of type f in D.

In Algorithm 2, if the sequence index of considered sequence $\overrightarrow{s}$ is greater than pruning threshold θ then $\overrightarrow{s}$ will be expanded with new event types. Additionally, considering $\overrightarrow{s}$ to be inserted into top-K sequences ranking, three scenarios are possible:

Algorithm 1 Procedure for discovering top-K sequential patterns

Require: D - dataset containing event types and their instances, F - set of event types.
Ensure: A set of top-K sequential patterns.
1: **for** each event type $f \in F$ **do**
2: Create 1-length sequence $\overrightarrow{s}$ from f.
3: TailEventSet($\overrightarrow{s}$) := $D(f)$.
4: *ExpandSequence($\overrightarrow{s}$).*
5: **end for**

1) If the length of the sequence $\overrightarrow{s}$ is at least *min_len*, and if there are few than K - 1 patterns in the top-K set, then $\overrightarrow{s}$ is inserted into the set (case 1 in Fig. 3).
2) If the length of the sequence $\overrightarrow{s}$ is at least *min_len* and there are K - 1 patterns in the top-K set, then $\overrightarrow{s}$ is inserted into the set and pruning threshold θ is set to sequence index of already K-th sequence in the set (case 2 in Fig. 3).
3) If the length of the sequence $\overrightarrow{s}$ is at least *min_len* and there are K patterns in the top set, then if the sequence index of $\overrightarrow{s}$ is equal to threshold theta θ, then $\overrightarrow{s}$ is inserted into top set (cases 5, 6 in Fig. 3).
4) If the length of the sequence $\overrightarrow{s}$ is at least *min_len* and there are K patterns in the top set, then if the sequence index of $\overrightarrow{s}$ is less than threshold theta θ, then $\overrightarrow{s}$ is not inserted into the set (case 3).
5) If the length of the sequence $\overrightarrow{s}$ is at least *min_len* and there are K patterns in the top set, then if the sequence index of $\overrightarrow{s}$ is greater than threshold theta θ, then $\overrightarrow{s}$ is inserted into the set, θ is set to the value of any K-th sequences' sequence index and all the sequences with sequence indexes less than θ are deleted from the top set (case 4).

In Fig. 3, we show possible scenarios where $\overrightarrow{s}$ is considered to be inserted into top-K sequences set.

In Algorithm 2, Spatial Join procedure performed in step 2 calculates a join set between tail event set of $\overrightarrow{s}$ and set of instances $D(f)$ (that is, calculates tailEventSet($\overrightarrow{s} \rightarrow f$)). Spatial join may be performed using the *plane sweep* algorithm proposed in [24]. Algorithm 3 calculates actual sequence index of sequence $\overrightarrow{s} \rightarrow f$. DR($\overrightarrow{s}[m] \rightarrow f$) in step 1 of Algorithm 3 is calculated as follows. The nominator of ratio in Definition 4 is the average density of instances from tailEventSet($\overrightarrow{s} \rightarrow f$)) inside the neighborhood spaces created for instances from tailEventSet($\overrightarrow{s}$). The denominator is the density of instances of type f (that is, $D(f)$) inside embedding space V.

V. Experiments

We performed experiments on both generated (synthetic) and real datasets. Our experiments have been conducted using machine with Intel Core i7-6700HQ CPU, each 2.6GHz and 16GB of RAM.

Algorithm 2 ExpandSequence($\overrightarrow{s}$)

Require: $\overrightarrow{s}$ - sequence to be expanded, K - number of top sequences to discover, min_len - minimal length of discovered sequences, θ - pruning threshold for top sequences.

for each event type $f \in F$ **do**
  TailEventSet($\overrightarrow{s} \rightarrow f$) := SpatialJoin(TailEventSet($\overrightarrow{s}$), $D(f)$).
  Calculate SequenceIndex($\overrightarrow{s} \rightarrow f$).
  if SI($\overrightarrow{s} \rightarrow f$) $\geq \theta$ **then**
    if length($\overrightarrow{s} \rightarrow f$) $\geq min\_len$ **then**
      if Number of already discovered sequences $<$ K - 1 **then**
        Insert $\overrightarrow{s}$ into the top sequences set.
      else if Number of already discovered sequences $=$ K - 1 **then**
        Insert $\overrightarrow{s}$ into the top sequences set.
        θ := sequence index of the actual K-th sequence in the top-K set.
      else
        Insert $\overrightarrow{s}$ into the top sequences set.
        if $SI(\overrightarrow{s}) > \theta$ **then**
          θ := sequence index of the actual K-th sequence in the top-K set.
          Delete all sequences from the top sequences set with the sequence indexes less than θ.
        end if
      end if
    end if
    ExpandSequence($\overrightarrow{s} \rightarrow f$).
  end if
end for

Algorithm 3 Calculate SequenceIndex($\overrightarrow{s} \rightarrow f$)

Require: $\overrightarrow{s} \rightarrow f$ - a sequence of event types; $\overrightarrow{s}[m]$ - the last event type participating in $\overrightarrow{s}$.
Ensure: Actual sequence index $SI(\overrightarrow{s} \rightarrow f)$.
return min(SI($\overrightarrow{s}$), DR($\overrightarrow{s}[m] \rightarrow f$)).

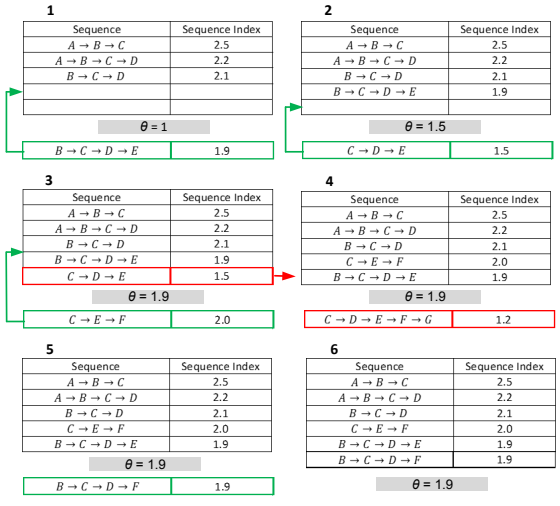


Fig. 3. Possibilities when $\overrightarrow{s}$ is considered to be inserted into top-K set with parameters min_len = 3 and K = 5

A. Experimental Results using Generated Data

We used the similar generator and notation of datasets names as proposed in [2]. In Table II, we recall parameters of data generator. In our experiments, we use cylindrical spatio-temporal neighborhood spaces $V_{N(e)}$ with parameters $R = 10$ (size of spatial dimension) and $T = 10$ (size of temporal window), similar to this one shown in Fig. 2. The whole spatio-temporal space V is given by parameters $DSize = 1000$ and $TSize = 1200$ (that is, both spatial dimensions are of size 1000 and temporal dimension is of size 1200). The total number of event instances in the dataset may be calculates as follows: $Pn * Ps * Ni * 2$, as in addition to patterns placed in a dataset we generate the same number of noise events.

We generated the same types of datasets as used in [2]. In Fig. 5, we show average computation times (we generated each dataset five times and averaged results) for three different types of datasets. In each case, computation time increases with increasing size of the dataset. We executed our algorithm for five values of K parameter (equal to 20, 40, 60, 80 and 100) and constant parameter min_len equal to 3. In Fig. 5, we are showing comparison of calculation time and the average number of discovered sequences for both STMiner proposed in [2] and our modification discovering top sequences set. The

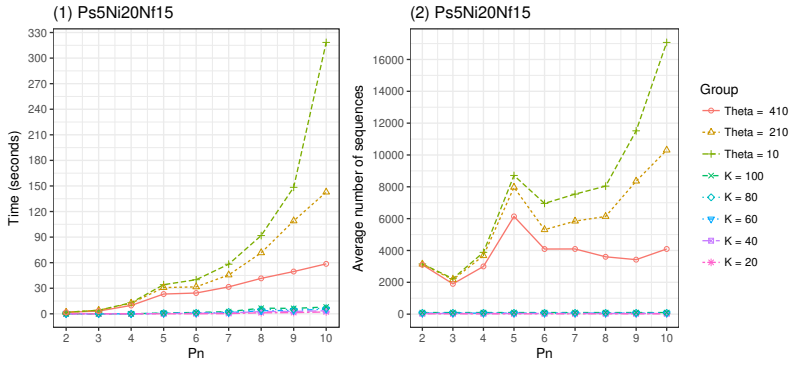


Fig. 4. The average computation times (plot 1) and average number of discovered sequences (plot 2) for both original STMiner algorithm proposed in [2] and our algorithm discovering top sequences set. The threshold $theta$ for STMiner has been set to three values: $10, 210, 410$

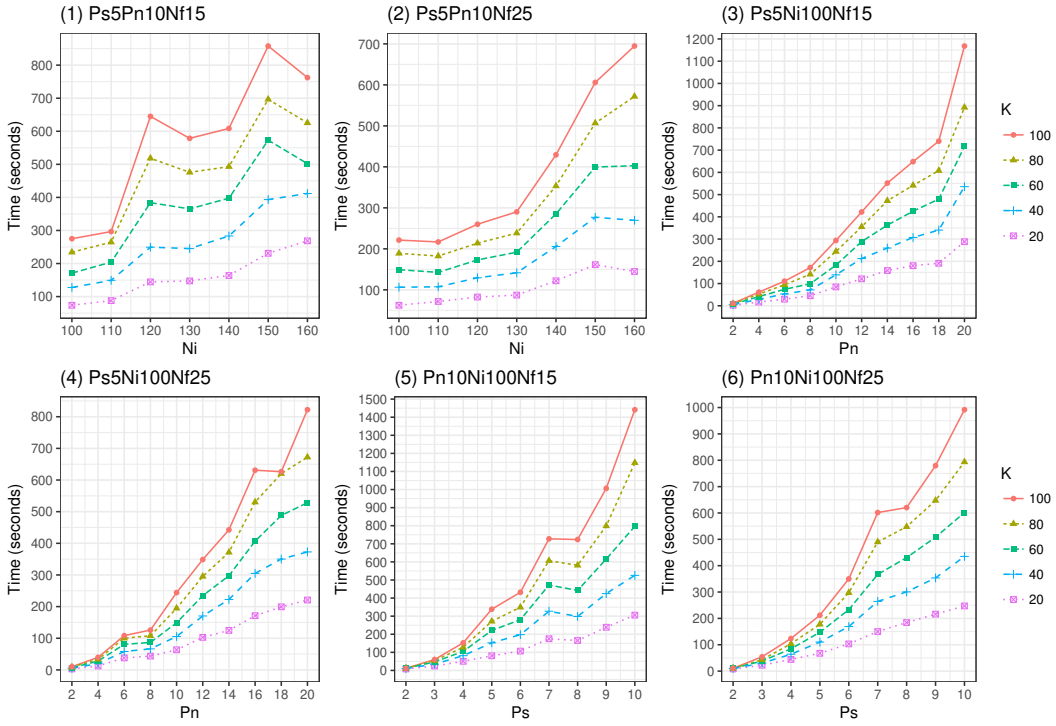


Fig. 5. Average computation times for randomly generated datasets with different number of instances per event type (diagrams (1), (2)), number of patterns (diagrams (3), (4)) and patterns lengths (diagrams (5), (6))

TABLE II
DESCRIPTION OF DATA GENERATOR PARAMETERS (ACCORDING TO [2])

Name	Description
Ps	Length (number of event types) of generated sequence
Pn	Number of sequences in generated data
$DSize$	Size of spatial dimensions of embedding space V
$TSize$	Size of temporal dimension of embedding space V
Nf	Total number of event types occurring in dataset
Ni	Number of instances per event type per sequence
R	Size of spatial dim. of neighborhood space $V_{N(e)}$
T	Size of temporal dim. of neighborhood space $V_{N(e)}$

size of the dataset for parameters $Pn = 10, Ps = 5, Ni = 20Nf = 15$ is 2000 event instances. As we may infer from Fig. 4, STMiner is impractical for even small datasets as it has a tendency to generate a huge number of redundant patterns. In Fig. 6, we show average calculation times for both STMiner and TopSTMiner when calculating exactly top 100 sequences set. To discover such sequences in STMiner algorithm we started with rather small θ threshold for sequence indexes and by its iterative increasing we obtained the set of 100 sequences.

B. Experimental Results using Real Data

For the first experiment on real data, we used the dataset of 14 types of pollutants available on the Internet repository

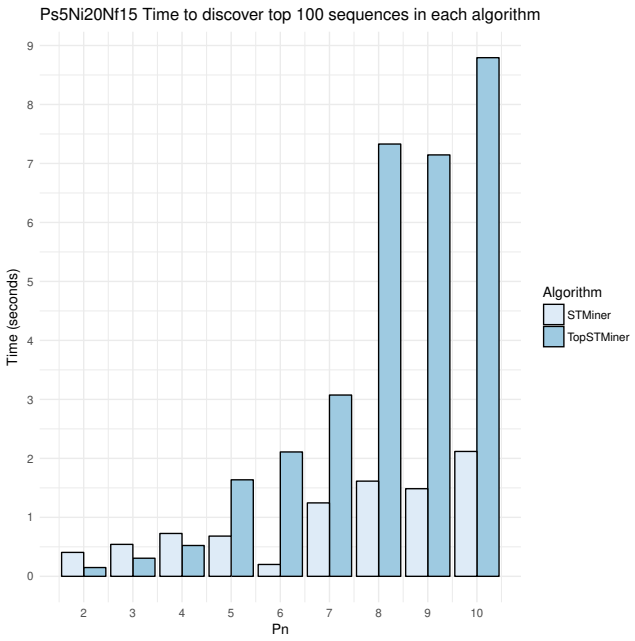


Fig. 6. The average computation times for both original STMiner algorithm proposed in [2] and our algorithm discovering top sequences set to calculate exactly top 100 sequences.

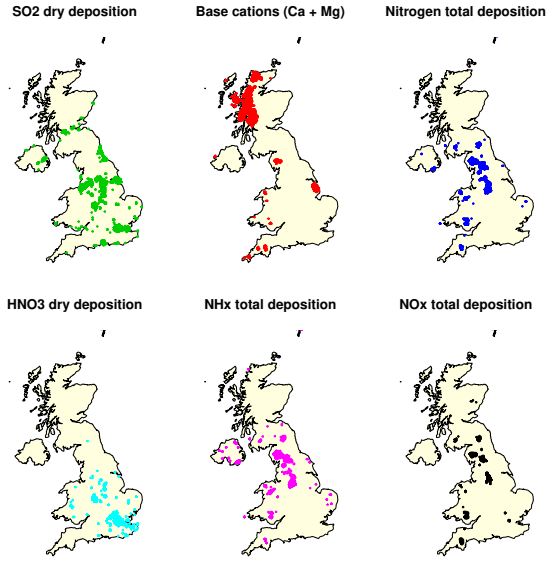


Fig. 7. Examples of extracted event types (SO2 dry deposition, base cations and total deposition of nitrogen, HNO3 dry deposition, NHx total deposition, NOx total deposition)

[25]. For each type of pollutant, the grids of resolution 5km^2 are available for years 2004-2014. Each grid contains numerical values of pollutant for United Kingdom region. For each grid and year, we calculated average value and standard deviation of pollutant. As abnormally high values of pollutants, we extracted events with values greater than three standard deviations from average. The task will be to investigate dependencies between these abnormal occurrences of pollutants. In Fig. 7, we show three types of events of pollutants extracted from the original dataset. We executed our algorithm with parameters $min\_len = 2$ and K $= 200$ and using cylindrical neighborhood space with parameters $R = 10$ km and $T = 1$ year. The types of pollutants available in the dataset and the number of abnormally high instances of each pollutant type in the final dataset are shown in Table III. In Table IV, we listed potentially interesting sequences form the top-100 set.

For the second experiment on real data we used the dataset of 6 pollutants obtained from 7 monitoring sites located in London Central: London Bloomsbury, London Eltham, London Haringey Priory Park South, London Harlington, London Hillingdon, London Marylebone, London Kensington. The name of pollutants and their numbers of instance in the extracted dataset are shown in Table V. The data have been obtained from the source [26]. Not each type of the pollutant is available for all of the stations. In Table VI, we show the name of each monitoring site, its location in the Northing, Easting system and available pollutants.

The original dataset contains hourly observations of pollutants shown in Table V for each day of 2015 for the stations mentioned above. For each type pollutant and for each station separately we extracted daily observations of such pollutant in the form of time series (that is, for each day we extracted 24 four observations respective to each hour). Then we clustered daily observations into four clusters to

TABLE III
TYPES OF POLLUTANTS USED IN THE FIRST EXPERIMENT (INST. - NUMBER OF INSTANCES)

Abbreviation	Pollutant type	Inst.
SOx-nss	Total deposition of oxidised sulphur	1989
SO4-nss	Wet deposition of sulphate	2256
SO2	Dry deposition of sulphur dioxide	1822
N	Total deposition of nitrogen	1339
NHx	Total deposition of reduced nitrogen	1252
NOx	Total deposition of oxidised nitrogen	579
NH3	Dry deposition of ammonia	1162
NH3-c	Concentration of ammonia	1292
NH4	Wet deposition of ammonium	2162
NO2	Dry deposition of nitrogen dioxide	698
HNO3	Dry deposition of nitric acid	1406
HNO3-c	Concentration of nitric acid	32
NO3	Wet deposition of nitrate	2021
Ca+Mg	Total deposition of base cations	2670
Ac	Total deposition of acidity	1406

TABLE IV
EXAMPLES OF PATTERNS DISCOVERED IN TOP-100 SET FOR REAL DATA FOR THE FIRST EXPERIMENT

Sequence	Sequence index
HNO3 $\rightarrow$ NO2	68.02
NO2 $\rightarrow$ HNO3	65.849
N $\rightarrow$ NOx $\rightarrow$ Ac $\rightarrow$ NHx $\rightarrow$ SOx	49.057
NOx $\rightarrow$ Ac $\rightarrow$ NHx $\rightarrow$ SO4	49.0339
NOx $\rightarrow$ Ac $\rightarrow$ NHx $\rightarrow$ NO3	47.8773
NOx $\rightarrow$ Ac $\rightarrow$ N $\rightarrow$ NH4	47.6831

TABLE V
TYPES OF POLLUTANTS USED IN THE SECOND EXPERIMENT

Pollutant type	Number of instances in dataset
Carbon Monoxide	52
Nitric Oxide	197
Nitrogen Dioxide	490
Ozone	534
PM10 particle deposition	161
PM2.5 particle deposition	73

obtain days with high concentration of the pollutant. For the clustering process we used R software, dtwclust package and distance time warping similarity between time series measure. The example of discovered clusters for Nitric Oxide pollutant for the London Eltham Station is shown in Fig. 8 and PM2.5 pollutant for the London Marylebone Station in Fig. 9. As the days with high concentration of pollutant we extracted these from cluster 2 for the former and cluster 4 for the latter. Each day with high pollutants' concentration has been marked as an event instance with event type corresponding to the pollutant type. The spatial location of the event instance is the location of respective monitoring station and the occurrence time is the corresponding day of occurrence.

We employed our algorithm to such dataset with parameters: $K = 100$, $min\_len = 2$, $R = 200$ meters and $T = 10$ days. The sizes of spatiotemoral space are as follows: $DSize1 = 37040$ meters, $DSize2 = 14262$ meters and $TSize$ = 364 days and are bounded by the locations of monitoring site and period of observation. The coordinates of stations are given in the Northing, Easting system. The parameter R specified as above means, that the algorithm will be looking for the interesting sequences considering events in each station separately. The set of top-15 sequences discovered from such dataset is shown in Table VII.

C. Results Discussion

For the experiments on synthetic data we show that even for small datasets our improvement discovering top sequences is more effective than the original algorithm STMiner proposed in [27]. As it has been explained, for many datasets and specific applications it may be difficult to provide a minimal sequence index threshold for discovered sequences. The algorithm proposed in the paper allows to eliminate this drawback by specifying the number of top sequences to discover. For the experimental results on real data we used two datasets, which have been preprocessed to obtain a set of event instances. For each of these datasets we obtain some potentially interesting sequences, however the additional usefulness of the proposed algorithm may be verified in the future experiments.

VI. CONCLUSIONS

In the paper, we consider the problem of effective discovering of top-K sequential patterns in event-based spatio-temporal data. In particular, we introduced the notion of top-K sequence (sequential pattern), we proposed the method creating set of top-K sequences and dynamically updating the set based on the rank of already expanded pattern. The approach allows to immediately prune patterns which for sure will not be

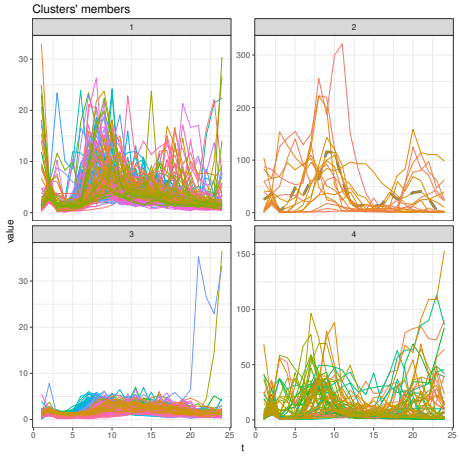


Fig. 8. Discovered clusters for Nitric Oxide pollutant for the London Eltham Station (cluster 2 contains days with high concentration of the pollutant)

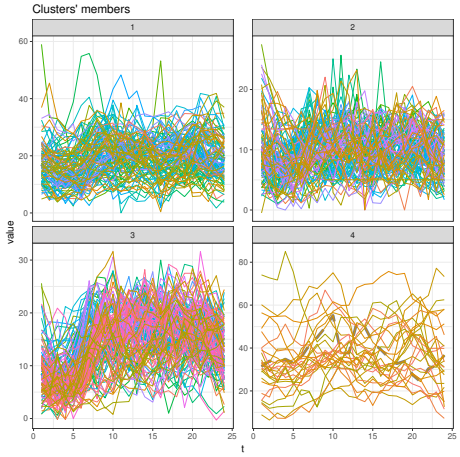


Fig. 9. Discovered clusters for PM2.5 pollutant for the London Marylebone Station (cluster 4 contains days with high concentration of the pollutant)

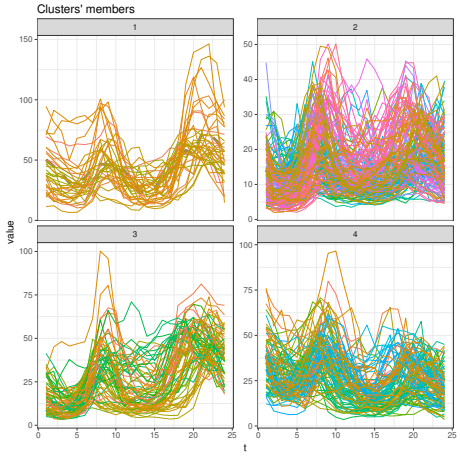


Fig. 10. Discovered clusters for Nitrogen Dioxide pollutant for the London Haringey Station (cluster 1 contains days with high concentration of the pollutant)

TABLE VI
MONITORING STATIONS, THEIR LOCATIONS (IN THE NORTHING, EASTING SYSTEM) AND AVAILABLE POLLUTANTS

Monitoring station	Location	Available pollutants
London Bloomsbury	530119, 182039	Nitric Oxide, Nitrogen Dioxide, Ozone, PM10, PM2.5
London Eltham	543981, 174655	Nitric Oxide, Nitrogen Dioxide, Ozone
London Haringey Priory Park South	529987, 188917	Nitric Oxide, Nitrogen Dioxide, Ozone, PM10, PM2.5
London Harlington	508295, 177800	Nitric Oxide, Nitrogen Dioxide, Ozone, PM10, PM2.5
London Hillingdon	506941, 178610	Nitric Oxide, Nitrogen Dioxide, Ozone
London Marylebone	528126, 182015	Carbon Monoxide, Nitric Oxide, Nitrogen Dioxide, Ozone, PM10, PM2.5
London Kensington	524045, 181749	Carbon Monoxide, Nitric Oxide, Nitrogen Dioxide, Ozone PM10, PM2.5

TABLE VII
EXAMPLES OF PATTERNS DISCOVERED IN TOP-100 SET FOR REAL DATA FOR THE SECOND EXPERIMENT

Sequence	Sequence index
PM10 → PM25	1000
PM25→ CarbonMonoxide	1000
PM25→ PM10	1000
PM25→ CarbonMonoxide→ NitrogenDioxide	974.079
CarbonMonoxide→ PM25	927.609
CarbonMonoxide→ PM25→ NitrogenDioxide	865.219
NitricOxide→ PM25	841.01
NitricOxide→ PM25→ PM10	841.01
CarbonMonoxide→ PM10	804.612
CarbonMonoxide → PM10 →PM25	804.612
CarbonMonoxide → PM10 → PM25 → Ni.Di.	804.612
NitrogenDioxide→ PM25	800.361
NitrogenDioxide→ PM25→ CarbonMonoxide	800.361
NitrogenDioxide→ PM25→ PM10	800.361
NitricOxide → PM25 → CarbonMonoxide	785.106

among the top-K sequences with length defined by *min_len* parameter. In the experiments, we show the efficiency of proposed approach. We also presented experimental results for real datasets. Obtained results are encouraging to investigate the topic in future research.

ACKNOWLEDGMENT

We acknowledge use of the dataset of UK Pollutants [25] available on the webpages http://www.pollutantdeposition.ceh.ac.uk/data and https://uk-air.defra.gov.uk/data/

REFERENCES

[1] Z. Li, *Spatiotemporal Pattern Mining: Algorithms and Applications*. Cham: Springer International Publishing, 2014, pp. 283–306.

[2] Y. Huang, L. Zhang, and P. Zhang, "A framework for mining sequential patterns from spatio-temporal event data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 4, pp. 433–448, April 2008.

[3] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining top-k frequent closed patterns without minimum support," in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, 2002, pp. 211–218.

[4] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53–87, Jan 2004. [Online]. Available: http://dx.doi.org/10.1023/B:DAMI.0000005258.31418.83

[5] P. Tzvetkov, X. Yan, and J. Han, "Tsp: mining top-k closed sequential patterns," in *Third IEEE International Conference on Data Mining*, Nov 2003, pp. 347–354.

[6] X. Yan, J. Han, and R. Afshar, *CloSpan: Mining: Closed Sequential Patterns in Large Datasets*, 2003, pp. 166–177. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611972733.15

[7] P. Terlecki and K. Walczak, "Efficient discovery of top-k minimal jumping emerging patterns," in *Rough Sets and Current Trends in Computing: 6th International Conference, RSCTC 2008 Akron, OH, USA, October 23-25, Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 438–447.

[8] R. Srikant and R. Agrawal, *Mining sequential patterns: Generalizations and performance improvements*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 1–17. [Online]. Available: http://dx.doi.org/10.1007/BFb0014140

[9] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, Mar 1995, pp. 3–14.

[10] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Machine Learning*, vol. 42, no. 1, pp. 31–60, Jan 2001. [Online]. Available: https://doi.org/10.1023/A:1007652502315

[11] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.

[12] C. H. Mooney and J. F. Roddick, "Sequential pattern mining – approaches and algorithms," *ACM Comput. Surv.*, vol. 45, no. 2, pp. 19:1–19:39, Mar. 2013. [Online]. Available: http://doi.acm.org/10.1145/2431211.2431218

[13] C. W. Wu, B.-E. Shie, V. S. Tseng, and P. S. Yu, "Mining top-k high utility itemsets," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 78–86. [Online]. Available: http://doi.acm.org/10.1145/2339530.2339546

[14] J. Yin, Z. Zheng, L. Cao, Y. Song, and W. Wei, "Efficiently mining top-k high utility sequential patterns," in *2013 IEEE 13th International Conference on Data Mining*, Dec 2013, pp. 1259–1264.

[15] F. Petitjean, T. Li, N. Tatti, and G. I. Webb, "Skopus: Mining top-k sequential patterns under leverage," *Data Min. Knowl. Discov.*, vol. 30, no. 5, pp. 1086–1111, Sep. 2016. [Online]. Available: http://dx.doi.org/10.1007/s10618-016-0467-9

[16] S. Shekhar, M. R. Evans, J. M. Kang, and P. Mohan, "Identifying patterns in spatial information: A survey of methods," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 193–214, 6 2011.

[17] P. Mohan, S. Shekhar, J. A. Shine, and J. P. Rogers, "Cascading spatio-temporal pattern discovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 11, pp. 1977–1992, Nov 2012.

[18] C. H. Yu, W. Ding, M. Morabito, and P. Chen, "Hierarchical spatio-temporal pattern discovery and predictive modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 979–993, April 2016.

[19] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung, "Mining, indexing, and querying historical spatiotemporal data," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: ACM, 2004, pp. 236–245. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014080

[20] Z. Li, B. Ding, J. Han, and R. Kays, "Swarm: Mining relaxed temporal moving object clusters," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 723–734, Sep. 2010. [Online]. Available: http://dx.doi.org/10.14778/1920841.1920934

[21] Z. Li, J. Wang, and J. Han, "eperiodicity: Mining event periodicity from incomplete observations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1219–1232, May 2015.

[22] P. Yin, M. Ye, W.-C. Lee, and Z. Li, "Mining gps data for trajectory

recommendation," in *Advances in Knowledge Discovery and Data Mining*, V. S. Tseng, T. B. Ho, Z.-H. Zhou, A. L. P. Chen, and H.-Y. Kao, Eds. Cham: Springer International Publishing, 2014, pp. 50–61.

[23] Y. Li, J. Bailey, L. Kulik, and J. Pei, "Mining probabilistic frequent spatio-temporal sequential patterns with gap constraints from uncertain databases," in *2013 IEEE 13th International Conference on Data Mining*, Dec 2013, pp. 448–457.

[24] L. Arge, O. Procopiuc, S. Ramaswamy, T. Suel, and J. S. Vitter, "Scalable sweeping-based spatial join," in *Proceedings of the 24rd International Conference on Very Large Data Bases*, ser. VLDB '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 570–581. [Online]. Available: http://dl.acm.org/citation.cfm?id=645924.671340

[25] Uk pollutant deposition data. [Online]. Available: http://www.pollutantdeposition.ceh.ac.uk/data

[26] Department for environment food and rural affairs archive data. [Online]. Available: https://uk-air.defra.gov.uk/data/

[27] M. R. Haylock, N. Hofstra, A. M. G. Klein Tank, E. J. Klok, P. D. Jones, and M. New, "A european daily high-resolution gridded data set of surface temperature and precipitation for 1950–2006," *Journal of Geophysical Research: Atmospheres*, vol. 113, no. D20, pp. n/a–n/a, 2008, d20119. [Online]. Available: http://dx.doi.org/10.1029/2008JD010201