Reducing and Linking Spatio-Temporal Datasets with kD-STR

Liam Steadman
l.steadman@warwick.ac.uk
University of Warwick
Coventry, United Kingdom

Nathan Griffiths
nathan.griffiths@warwick.ac.uk
University of Warwick
Coventry, United Kingdom

Stephen Jarvis
s.a.jarvis@warwick.ac.uk
University of Warwick
Coventry, United Kingdom

Mark Bell
mbell@trl.co.uk
TRL
Wokingham, United Kingdom

Shaun Helman
shelman@trl.co.uk
TRL
Wokingham, United Kingdom

Caroline Wallbank
cwallbank@trl.co.uk
TRL
Wokingham, United Kingdom

ABSTRACT

When linking spatio-temporal datasets, the kD-STR algorithm can be used to reduce the datasets and speed up the linking process. However, kD-STR can sacrifice accuracy in the linked dataset whilst retaining unnecessary information. To overcome this, we propose a preprocessing step that removes unnecessary information and an alternative heuristic for kD-STR that prioritises accuracy in the linked output. These are evaluated in a case study linking a road condition dataset with air temperature, rainfall and road traffic data. In this case study, we found the alternative heuristic achieved a 19% improvement in mean error for the linked air temperature features and an 18% reduction in storage used for the rainfall dataset compared to the original kD-STR heuristic. The results in this paper support our hypothesis that, at worse, our alternative heuristic will yield a similar error and storage overhead for linking scenarios as the original kD-STR heuristic. However, in some cases it can give a reduction that is more accurate when linking the datasets whilst using less storage than the original kD-STR algorithm.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; Decision support systems; Data mining.

KEYWORDS

Spatio-temporal data, Data reduction, Partitioning, Modelling, kD-STR

ACM Reference Format:
Liam Steadman, Nathan Griffiths, Stephen Jarvis, Mark Bell, Shaun Helman, and Caroline Wallbank. 2020. Reducing and Linking Spatio-Temporal Datasets with kD-STR. In *3rd ACM SIGSPATIAL Workshop on Advances in Resilient and Intelligent Cities (ARIC'20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3423455.3430317

ARIC'20, November 3–6, 2020, Seattle, WA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8165-9/20/11...$15.00
https://doi.org/10.1145/3423455.3430317

1 INTRODUCTION

The growth of smart and connected cities has introduced a range of new urban spatio-temporal datasets that are voluminous, varied and contain significant autocorrelation. Often, data scientists wish to analyse multiple spatio-temporal datasets in the context of each other. In some analyses, co-occurrence mining is used to detect events that occur at the same time and location from different datasets [11]. In other scenarios, *supplementary* datasets are used to provide context and information about the instances in a *primary* dataset. For example, Ding *et al.* used weather and demographic data to augment traffic data in Shanghai [9], and Knittel *et al.* augmented a mortality dataset with information from pollution and other datasets [10]. This process is referred to as dataset *linking* or *augmenting*, and is a common procedure when analysing datasets.

However, in recent years the volume of spatio-temporal datasets has increased significantly making the linking process computationally expensive or infeasible. This growth has been driven by new and cheaper sensors, as well as the growth of urban populations.s To tackle this problem, methods exist to reduce the quantity of data that needs to be processed during linking. Amongst these, the kD-STR algorithm can be used to reduce the quantity of data in the datasets whilst minimising the information lost [17, 18]. However, whilst kD-STR can be used to reduce the datasets, its aim is to minimise the error incurred in the original datasets. In the context of linking datasets, it may be more beneficial to *minimise the difference between the features engineered using the raw and reduced datasets.* Furthermore, kD-STR fails to consider the characteristics of the primary dataset when reducing the supplementary datasets. This can lead to a less efficient reduction of the supplementary datasets and less accurate feature engineering during the linking process than could otherwise be achieved.

Therefore, the aim of this research is to decrease the error incurred when augmenting a primary dataset using supplementary datasets that have been reduced with kD-STR. More precisely, in this paper, we present three contributions:

(1) We discuss how a more efficient reduction of supplementary datasets using kD-STR can be achieved by considering the spatial and temporal characteristics of the primary dataset.
(2) We present an alternative kD-STR heuristic for reducing supplementary datasets that improves accuracy in linking scenarios. This heuristic prioritises information retention in areas and time periods of the supplementary dataset that are applicable to the primary dataset.

(3) We demonstrate the utility of the alternative heuristic in a case study which links road condition data with road traffic, air temperature and rainfall datasets.

The remainder of this paper is structured as follows. In Section 2, we present methods, including kD-STR, for linking spatio-temporal datasets and reducing the storage overheads of such datasets. In Section 3, we present the preprocessing step and alternative heuristic proposed in this paper. In Sections 4 and 5 we present an empirical study demonstrating the utility of this alternative heuristic. Finally, in Sections 6 and 7 we discuss the impact of these results.

2 BACKGROUND

When linking datasets, the quantity of data can slow the process significantly, often making the process infeasible. In this section, we first discuss common methods for augmenting one dataset with information calculated from one or more supplementary datasets. Second, we discuss the kD-STR algorithm for reducing the quantity of data stored in a dataset whilst minimising the information lost.

2.1 Linking Datasets

To augment a primary spatio-temporal dataset with information from a supplementary spatio-temporal dataset, linking methods aim to estimate the supplementary feature values at the time and location appropriate for each primary instance[1]. For each primary instance, one or more supplementary instances are selected and features engineered over them. One simple but often-used method is to select all supplementary instances within a fixed spatio-temporal radius of the primary instance. This is referred to as *neighbourhood* linking, and has been used in projects such as the Australian Urban Research Infrastructure Network (AURIN) [16]. Neighbourhood linking allows for different distance metrics to be used, such as Euclidean distance, and allows for either separate spatial and temporal distance limits or a combined spatio-temporal distance limit. In the latter case, the spatial and temporal distance metrics are combined by use of a weighting factor [6]. This allows for supplementary instances that are spatially close but distant in time, and instances that are spatially far apart but close in time, to be combined for a more accurate estimate of the supplementary feature values. Other examples of neighbourhood methods, including inverse distance weighting of instances, can be found in literature [5, 10].

However, when the distribution of supplementary instances in the spatial or temporal domains is highly varied, one primary instance may be linked to many supplementary instances while others are linked to few. When this is undesirable, nearest neighbour (NN) methods, which select a fixed number of nearest supplementary instances for each primary instance, are more appropriate. Again, a weighting between the spatial and temporal distance metrics is required to select the nearest supplementary instances to a primary instance [19, 20, 22, 23]. Other methods for linking spatio-temporal datasets include kriging and radial basis functions, as well as machine learning-based approaches [15]. In this paper, we use the NN and neighbourhood methods as examples of linking functions, though we assert this work is linking function agnostic.

2.2 Reducing Datasets

To speed up the process of linking large spatio-temporal datasets, data reduction techniques can be used. These aim to reduce the quantity of data stored in a dataset whilst minimising the information lost for a given scenario. Commonly used data reduction techniques include instance and feature selection/engineering techniques, such as Principle Components Analysis (PCA) [14]. These reduce the quantity of data to be processed by reducing the number of instances or features in the dataset. However, these techniques can remove important information from the supplementary datasets. Other algorithms for reducing spatio-temporal data include the IDEALEM [21] and ISABELA [12] algorithms, and a two-part algorithm proposed by Pan *et al.* [13]. A discussion of these algorithms can be found in literature [17].

One such reduction technique is the k-Dimensional Spatio-Temporal Reduction (kD-STR) algorithm, which takes advantage of the spatial and temporal autocorrelation present in spatio-temporal data [17, 18]. The kD-STR algorithm is iterative and partitions a dataset into partitions of similar instances, then replaces the instances within each partition with a model of their feature values. On each iteration, a heuristic function is used to decide between improving an existing model and increasing the number of partitions and models stored, with the choice that minimises the heuristic value being taken. The heuristic function used in kD-STR balances the error introduced by the reduction process with the decrease in storage used by the reduced dataset. The error introduced is measured by reconstructing the original instances from the models and measuring the difference between the original instances and their estimated values. However, when linking datasets, which is the focus of this paper, minimising the reconstruction error of the original dataset is less important than minimising the error incurred in the feature engineering step of the data linking process. Furthermore, kD-STR retains information about all instances during reduction, yet for data linking purposes this may be unnecessary. Addressing these two shortcomings is the focus of this paper.

3 DATA REDUCTION PROCESS FOR SUPPLEMENTARY DATASETS

In the context of linking datasets, kD-STR can be used to reduce supplementary datasets to speed up the linking process – a description and pseudocode of the kD-STR algorithm can be found in Appendix A. To augment the primary dataset using the reduced supplementary datasets, each primary instance is compared against the location and time of the partitions in the reduced datasets rather than each instance in the raw supplementary datasets. However, kD-STR fails to take advantage of the characteristics of the primary and supplementary datasets which may yield a more efficient reduction. To overcome this weakness, we propose two changes to the kD-STR reduction process: (i) a preprocessing step that reduces the spatial and temporal resolution of the datasets where possible and removes unnecessary information; (ii) an alternative kD-STR heuristic that minimises the error incurred in the features engineered during the linking process, and considers the spatio-temporal distribution of the primary instances. These two changes are discussed in Sections 3.1 and 3.2 respectively, and the overall reduction process can be seen in Algorithm 1.

[1] In many cases this is the time and location the primary instance was recorded at. However, it may be different – for example we may wish to know the feature values at the same location but 1 hour prior to the recorded time of the primary instance.

Algorithm 1: Reducing n supplementary datasets using knowledge of D_{prim}

```
Data: Primary dataset $D_{\text{prim}}$ and supplementary datasets
      $D_1, ..., D_n$
Result: The reduced datasets $\langle P_1, M_1 \rangle, ..., \langle P_n, M_n \rangle$
$r_s \leftarrow$ spatialRes($D_{\text{prim}}$)
$r_t \leftarrow$ temporalRes($D_{\text{prim}}$)
$D_{\text{prim}'} \leftarrow$ sample($D_{\text{prim}}, r_s, r_t$)
for $1 \leq i \leq n$ do
    $S^i_{\text{bound}} \leftarrow$ spatialBoundary($D_i, D_{\text{prim}}$)
    $T^i_{\text{bound}} \leftarrow$ temporalBoundary($D_i, D_{\text{prim}}$)
    $D_i \leftarrow$ removeOutside($D_i, S^i_{\text{bound}}, T^i_{\text{bound}}$)
    $D_i \leftarrow$ reduceRes($D_i, r_s, r_t$)
    $\langle P_i, M_i \rangle \leftarrow$ alternativeKDSTR($D_i, D_{\text{prim}'}$)
end
```

3.1 Pre-Processing Resolution Reduction

Given a primary dataset D_{prim} and n supplementary datasets $D_1, ..., D_n$, our aim is to remove any information in $D_1, ..., D_n$ that is known to be irrelevant or unnecessary for D_{prim}. To achieve this, we introduce a preprocessing step consisting of two parts to be used prior to kD-STR. First, unnecessary instances are removed. For a supplementary dataset D_i, where $1 \leq i \leq n$, a spatial boundary $S^i_{\text{bound}} = \langle x_b, x_e, y_b, y_e \rangle$ and temporal boundary $T^i_{\text{bound}} = \langle t_b, t_e \rangle$ are computed, where S is the spatial domain and T is the temporal domain. Forming the boundaries requires knowledge of the instances in D_{prim} and D_i, as well as knowledge of the furthest a supplementary instance in D_i can be to remain applicable to an instance in D_{prim}. This knowledge may come from the spatial and temporal distributions of the datasets (using measures such as the maximum distance between any instance D_{prim} and its nearest neighbour in D_i or experimental variograms of the datasets), or prior knowledge provided by the user. After the spatial and temporal boundaries are computed, all instances in D_i that are located outside of S^i_{bound} and T^i_{bound} are removed.

The second part of the preprocessing step removes unnecessary resolution in D_i. Consider a case in which the maximum distance between any sensor in D_i is 1.5 miles, and each sensor records an instance every 15 minutes. If each sensor in D_{prim} is 20 miles apart from its neighbours and records one instance per day, and we wish to augment each instance $d_{s,t} \in D_{\text{prim}}$ with the daily mean value recorded at the nearest sensor to s, the resolution of D_i is unnecessarily high. We can therefore reduce the resolution by removing some sensors in D_i and aggregating the instances at each sensor in D_i to daily mean values.

After removing unnecessary information, each supplementary dataset D_i is reduced using the kD-STR algorithm with an alternative heuristic function as explained in Section 3.2.

3.2 kD-STR with Alternative Heuristic Function

After removing the unnecessary information from each supplementary dataset, the reduction process reduces each dataset using kD-STR. However, the aim of reducing supplementary datasets is to minimise the error incurred in the features engineered during the linking process. This may not be the same as minimising the error incurred in the original supplementary features. Thus, while the heuristic function used in kD-STR minimises the information lost in the original supplementary dataset, it ignores the error incurred in the features engineered during linking. To overcome this, the alternative heuristic function shown in Equation 1 is proposed.

$$h(D_{\text{prim}'}, D_i, \langle P_i, M_i \rangle) = \alpha \cdot q(D_i, \langle P_i, M_i \rangle) + (1 - \alpha) \cdot e(D_{\text{prim}'}, \langle P_i, M_i \rangle) \quad (1)$$

Here, $q(D_i, \langle P_i, M_i \rangle)$ measures the ratio between the volume of data required to store the reduced dataset and that required to store the original dataset. Furthermore, $e(D_{\text{prim}'}, \langle P_i, M_i \rangle)$ measures the difference between the features engineered during the linking process using the raw supplementary dataset D_i versus those engineered using the reduced supplementary dataset $\langle P_i, M_i \rangle$. The parameter α is a weighting factor that balances the user's desire for minimised storage versus minimised error. This value is bounded to the range $[0, 1]$ and must be chosen before reducing the dataset. Values of α close to 0 minimise error at the cost of increased storage overhead by reducing the weight of $q(D_i, \langle P_i, M_i \rangle)$. Similarly, values close to 1 prioritise storage reduction over incurred error by reducing the weight of $e(D_{\text{prim}'}, \langle P_i, M_i \rangle)$.

The functions used in Equation 1 are defined in Equations 2, 3 and 4. In Equation 3, $D'_{\text{prim}'}$ is a subset $D_{\text{prim}'}$ of the primary dataset after it has been augmented using the reduced supplementary dataset $\langle P_i, M_i \rangle$, $d^f_{s,t}$ is the value of feature f for instance $d_{s,t} \in D_{\text{prim}'}$, and LF_i is the set of features engineered using the raw supplementary dataset D_i. Finally, $d'^f_{t,s}$ is the value of the same feature engineered using the reduced supplementary dataset $\langle P_i, M_i \rangle$.

$$q(D_i, \langle P_i, M_i \rangle) = \frac{storage(\langle P_i, M_i \rangle)}{storage(D_i)} \quad (2)$$

$$e(D_{\text{prim}'}, \langle P_i, M_i \rangle) = \frac{1}{|LF_i|} \sum_{f \in LF_i} \frac{\psi(f, D_{\text{prim}'}, D'_{\text{prim}'})}{\text{range}(f)} \quad (3)$$

$$\psi(f, D_{\text{prim}'}, D'_{\text{prim}'}) = \sqrt{\frac{\sum_{d_{s,t} \in D_{\text{prim}'}} (d^f_{t,s} - d'^f_{s,t})^2}{|D_{\text{prim}'}|}} \quad (4)$$

To calculate the error incurred by augmenting the primary dataset with reduced supplementary datasets, the linking process for the entire primary dataset must be completed. This means the linking process we wish to speed up has to be completed at least once during the reduction process. To overcome this issue, a sample of the primary dataset that is representative of its spatial and temporal distributions is used, $D_{\text{prim}'} \subset D_{\text{prim}}$. Since primary instances that are close in space and time will be augmented by the same supplementary instances, a sample of the primary dataset that has the same distribution of instances in space and time is sufficient for estimating the error incurred in the engineered features.[2] Only partitions that contain an instance from $D_{\text{prim}'}$ within their spatio-temporal

[2] Note that the feature values of the primary dataset are not important when creating the sample, only the distribution of instance locations in space and time.

bounds will be able to store more than 1 model coefficient. Thus, a sample that maintains the spatio-temporal distribution is adequate. Using a sample that is not representative of the spatio-temporal distribution of D_{prim} would result in information retention in $\langle P_i, M_i \rangle$ being disproportionately focused in areas and time periods that are less relevant to the instances in D_{prim}.

One method for creating a representative sample that retains the relative distribution in space and time is to use a stratified sampling technique [8]. Alternatively, a sample of instances can be chosen at random. Then, each instance in the sample can be replaced if it is within γ_S spatial distance and γ_T temporal distance of its nearest neighbour in the sample [7]. An appropriate sample size or appropriate distances γ_S and γ_T would retain the distribution of instances in D_{prim} over space and time whilst minimising the number of instances in the sample to aid quick reduction.

The overall process of reducing the supplementary datasets is shown in Algorithm 1. Here, lines 1–3 calculate the spatial and temporal resolutions of the primary dataset, and create a sample of the dataset that is representative of its distribution in space and time. Furthermore, lines 5–7 calculate the spatial and temporal boundary of instances in D_i that are applicable to D_{prim}. Finally, line 8 reduces the spatial and temporal resolution of D_i given the resolution of D_{prim}, and line 9 performs kD-STR reduction on D_i using the alternative heuristic presented in Equation 1.

4 EXPERIMENTAL EVALUATION

To evaluate the impact of the alternative heuristic function, we augmented a road pavement condition dataset with information from air temperature, road traffic and rainfall datasets. Each instance in the primary dataset contained several features measured on two dates at a specific location. Therefore, the aim was to engineer features about the air temperature, traffic and rainfall that occurred at that location between the two dates defined for each instance. The three supplementary datasets are described in Section 4.1. To provide a baseline for comparison, we augmented the primary dataset using the raw supplementary datasets, and this process is described in Section 4.2. For the purpose of analysing where information was retained and lost, we did not perform the instance removal steps of the alternative process (lines 5–8 of Algorithm 1). Furthermore, we used the nearest neighbour and neighbourhood methods to evaluate the alternative heuristic, however we believe this work to be agnostic to the linking method used[3].

Our aim was to test three hypotheses:

H1 Compared to the linked baseline, the features engineered using the reduced datasets that were reduced using the alternative heuristic function will be more accurate than the datasets that were reduced using the original heuristic function. Since the alternative heuristic aims to minimise the error incurred in the linking process rather than the reconstruction of the original supplementary instances, this should be the case.

H2 Compared to the original heuristic function, the alternative heuristic function will prioritise information retention in those partitions nearest to the instances in D_i. That is, only partitions which contain the spatial locations of the instances in D_i and overlap with the time periods of each instance will store more than 1 model coefficient.

H3 When the original supplementary instances are reconstructed from the reduced dataset, those instances that are far from the locations and times of the primary instances will be less accurately reconstructed than those that are close to the primary instances. Since those supplementary instances that are far from the primary instances will not have been used in the linking process by the alternative heuristic function, the partitions in which they reside will not be modelled as accurately as those that are used in the linking process.

4.1 Datasets

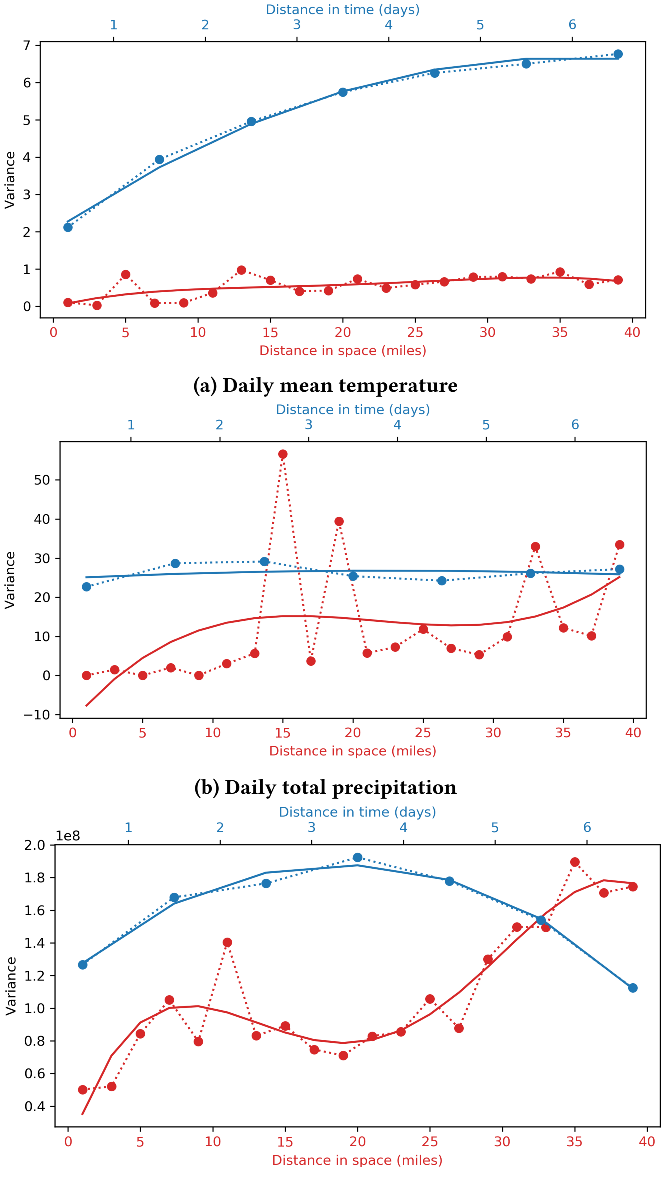
One primary dataset and three supplementary datasets were used in this experiment:

- **Road Pavement Condition (primary)**
  This dataset contained features about the condition of the M1 northbound motorway in England between June 2016 and September 2017, and contained 2719 instances [4]. Each instance related to a 10 meter section of road between a start and end date. The mid-point of the road segment was used as a single point to represent the location of the road section in space.
- **Air Temperature and Rainfall (supplementary)**
  These datasets were sourced from the Met Office Integrated Data Archive System [1, 2], and contained all instances recorded at 87 air temperature sensors and 69 rainfall sensors in England in 2016 and 2017. Experimental variograms of the daily mean air temperature and total daily precipitation features can be seen in Figures 1a and 1b. The two datasets contained 61,939 and 48,829 instances respectively.
- **Road Traffic (supplementary)**
  This dataset was sourced from the Highways England WebTRIS dataset [3]. It contained the total daily vehicle count and vehicle lengths at 155 sensors on the M1 northbound for each day in 2016 and 2017. For simplicity, we only considered sensors on the main carriageway and omitted sensors on entry and exit slip-roads. An experimental variogram of the total daily vehicle count feature can be seen in Figure 1c. The dataset contained 69,715 instances.

4.2 Linked Data Baseline

To create a baseline linked dataset, we augmented the instances in the primary dataset using the raw supplementary datasets. For each instance in the primary dataset, the mean daily min, mean and max air temperature, mean daily rainfall and mean daily traffic count were imputed at the location of the primary instance between the start and end date of that instance. To impute these values, we tested the neighbourhood and nearest neighbour (NN) methods, and their inverse distance weighted (IDW) variants, with a range of parameter values to see which parameter values yielded the most accurate imputations. This resulted in a neighbourhood of 40 miles and 0 days being used to impute the air temperature and rainfall for each primary instance when using the neighbourhood

[3] Equation 3 calculates the difference between the original and reconstructed features created by the linking method. Therefore, any linking method can be used as the error is calculated using the difference in imputed values.



(a) Daily mean temperature

(b) Daily total precipitation

(c) Daily total traffic count

Figure 1: Experimental variograms of the daily mean temperature, total precipitation and traffic count features from the air temperature, rainfall and traffic datasets.

and IDW neighbourhood methods, and neighbourhood of 20 miles for the traffic dataset. For the NN and IDW NN techniques, a value of $k = 5$ was found to be most accurate for the air temperature and traffic datasets, whilst a value of $k = 1$ was most accurate for the rainfall dataset. Using these techniques, a baseline for each of the four linking methods was created.

5 RESULTS

After reducing each of the supplementary datasets using kD-STR with the original and alternative heuristic functions, the primary dataset was augmented using the reduced datasets. For each primary instance, supplementary instances were imputed from each of the reduced datasets at the location of the primary instance for each day between the start and end dates of the primary instance.

Augmenting the primary dataset using the reduced supplementary datasets was found to be faster than using the raw supplementary datasets. Using the four baseline linking methods on the raw datasets described in Section 4.2, the linking process took between 8,067 and 13,334 seconds. In comparison, augmenting the primary dataset using the reduced datasets took between 308 and 329 seconds using kD-STR when $\alpha = 0.9$, a value which indicates a strong preference for reducing the quantity of data stored at the cost of reduced accuracy. When $\alpha = 0.01$, indicating a preference to retain accuracy at the cost of data storage, the linking took between 3,593 and 3,698 seconds. The time taken to link the reduced supplementary datasets for different values of α are shown in Figure 2. These tests were performed on an isolated workstation with an Intel i5-8700k and 48GB RAM, though only a single process was used.

In the remainder of this section we present the results of testing each of our three hypotheses. In Section 5.1 we discuss the error incurred in the linked features by reducing the supplementary datasets prior to linking, addressing hypothesis H1. In Section 5.2 we discuss the retention of information in space and time, addressing H2. Finally, in Section 5.3 we discuss the error incurred in the raw datasets, addressing H3.

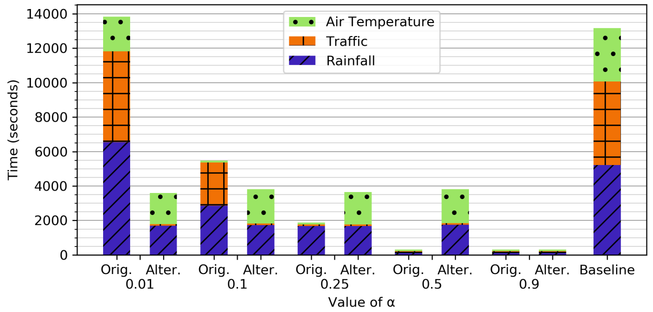


Figure 2: Time taken to link primary and supplementary datasets with NN method as baseline. Results are shown for the original and alternative heuristic functions.

5.1 Error incurred in the Linked Features

By reducing the supplementary datasets, the feature values engineered during linking were different to those engineered using the raw supplementary datasets. For example, Figure 3 shows the distribution of feature values for the mean temperature and total precipitation features. As hypothesised, we found that decreasing the value of α, indicating a stronger preference for information retention, increased the number of partitions and models used to store each of the reduced datasets. In all cases, this allowed the distribution of the features engineered using the reduced datasets to better match the distribution of features engineered using the raw dataset, either by more accurately matching the range of feature values observed or more closely matching the broad shape of the distribution.

In most cases, decreasing the value of α lead to a decrease in the maximum error between features engineered using the reduced and raw datasets. As shown in Figure 4, the more accurate distribution of engineered values is matched by a more accurate engineered value for each instance when $\alpha = 0.01$ compared to $\alpha = 0.9$. However, in many cases, such as the Min Temp and Max Temp features, the median and third quartile errors increased when $\alpha = 0.01$. This is indicative of the engineered feature values better matching for some instances but not all as the output models try to more accurately capture the nuances of the supplementary dataset features. We note that higher percentage error of the Total Precipitation feature is caused by the high spatial and temporal variance of the rainfall dataset compared to the lower error and lower spatial variance of the air temperature dataset. For the traffic dataset, high spatial and temporal variance of the data made increasing the number of partitions too costly in storage. Thus, when $\alpha = 0.01$, which indicates a strong preference for reduced error at the cost of increased storage volume, the number of partitions stored for the traffic dataset was still 1 and the error incurred was approximately the same as when $\alpha = 0.9$.

A comparison of the storage used by each of the three supplementary datasets and the error in features they are used to engineer in the linking process can be seen in Figure 5. Each subfigure shows the error versus storage for kD-STR using the original and alternative heuristics on each of the datasets. We have also split the results into those reductions that output 1 partition versus those that output more than 1 partition. As expected, the two heuristics were unable to prioritise information retention in the partitions most applicable to the primary dataset when the reduced datasets contained only 1 partition. However, when more than 1 partition was stored, the alternative heuristic was able to prioritise information retention in just those partitions that overlap with the areas and time periods that are applicable to the primary dataset. In our results, no reduced dataset that had been reduced using the original heuristic was able to achieve a lower error and lower storage cost than a dataset reduced using the alternative heuristic. In Section 5.2 we show further evidence that the alternative heuristic retained information only in the areas and time periods applicable to the primary dataset.

5.2 Retention of Information in Space and Time

To explore where in space and time kD-STR used models with a higher number of coefficients (and thus retained more information), we plotted the boundaries of the partitions that stored more than 1 model coefficient in the reduced datasets. Figure 6 shows the locations of these partitions in time when both heuristic functions were used to reduce the air temperature dataset. As shown, every partition output by kD-STR with the alternative heuristic overlapped with the time period for which instances exist in the primary dataset. That is, only partitions in the reduced dataset that were linked to the primary dataset during linking stored more than 1 model coefficient. In contrast, the results for the original heuristic function show that many partitions that were not applicable to the primary dataset have complex partitions, meaning kD-STR chose to retain information in time periods that were not useful to the primary dataset.

Similar conclusions were drawn from all three supplementary datasets. The low variation in the spatial domain of the air temperature dataset resulted in partitions that cover large spatial areas and short time periods. This meant it was likely any partition would overlap with the area that is applicable to the primary dataset. However, the high variation in the temporal domain resulted in partitions that only cover a small number of time intervals, giving the results shown in Figure 6. For the reduced rainfall and traffic datasets, the variance of the datasets in space and time resulted partitions with a larger range of spatial areas and time periods. Again, no partition stored more than 1 model coefficient unless it intersected the area and time period applicable to the primary dataset when reduced with the alternative heuristic, but several partitions that did not intersect stored more than 1 model coefficient when reduced with the original heuristic.

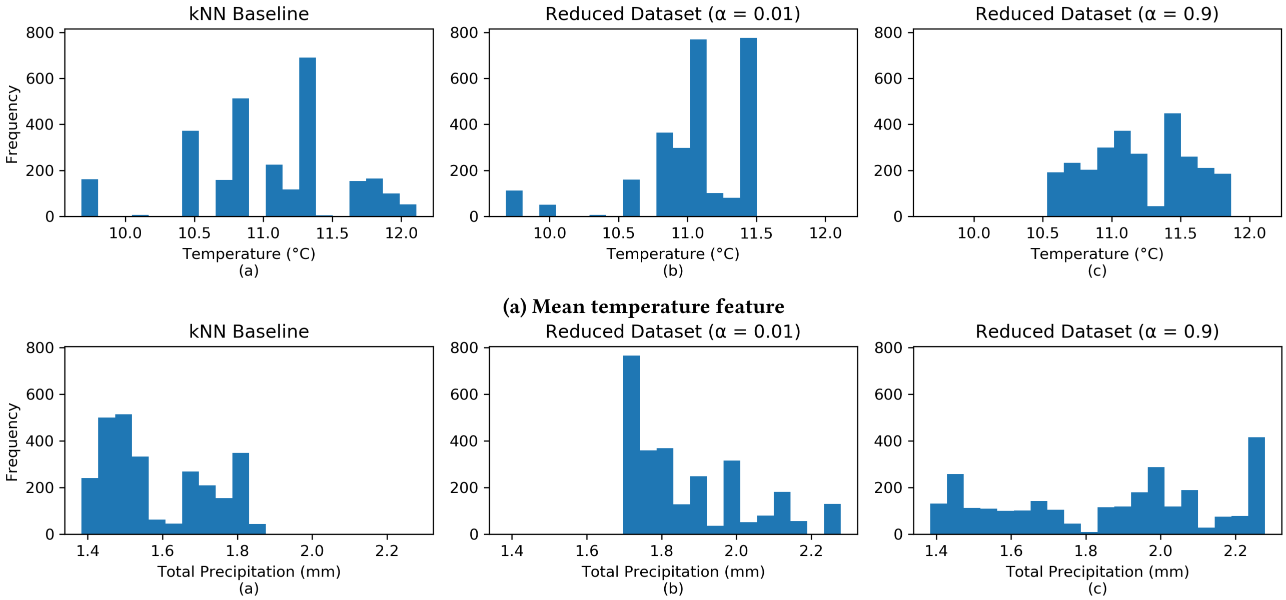
5.3 Error incurred in the Original Features

The error incurred by reducing the supplementary datasets using the alternative heuristic function was higher or approximately equal to reducing using the original heuristic function given approximately the same storage volume used. This result only occurred when the number of partitions store was above 1, as discussed in Section 5.1. Figure 7 shows the error incurred when the original supplementary instances were reconstructed from the reduced datasets versus the storage used by the reduced dataset. As shown, no reduced dataset that was reduced using the alternative heuristic achieved a more accurate reconstruction of the original instances whilst requiring less storage than a dataset reduced using the original heuristic. A more direct comparison for the mean temperature can be seen in Figure 8, where the error incurred by the original heuristic function was consistently less than or equal to the error incurred by the alternative heuristic function. Note that the exception of $\alpha = 0.5$ was caused by the alternative heuristic function using more than 1 partition whilst the original heuristic had increased the number of partitions used[4].

6 DISCUSSION

By reducing supplementary datasets using kD-STR, the dataset linking process can be sped up. Though the reduced datasets do not offer a perfect representation of the features engineered using the raw datasets, the accuracy achieved can be sufficient for many use cases given the speedup offered for linking tasks. However, by using the alternative heuristic described in this paper, a reduction can be achieved that yields similar or better accuracy of feature engineering in the linking process. More specifically, our key findings were: (i) the error incurred in the feature engineering step of data linking can be reduced by using our alternative heuristic; (ii) however, the reconstruction error of the original supplementary features can be worsened; (iii) when using the alternative heuristic, information retention is focused on those partitions that are used

[4] When reducing the supplementary datasets using kD-STR with the original heuristic function versus the alternative heuristic function, we found the relationship between α, the error metric and the storage metric was different.



(a) Mean temperature feature

(b) Total precipitation feature

Figure 3: Histograms of the mean temperature and total precipitation features engineered using the NN baseline method and raw supplementary datasets, and estimating at the road survey location using the reduced dataset, after reduction using the alternative heuristic function with $\alpha = 0.01$ and $\alpha = 0.9$.

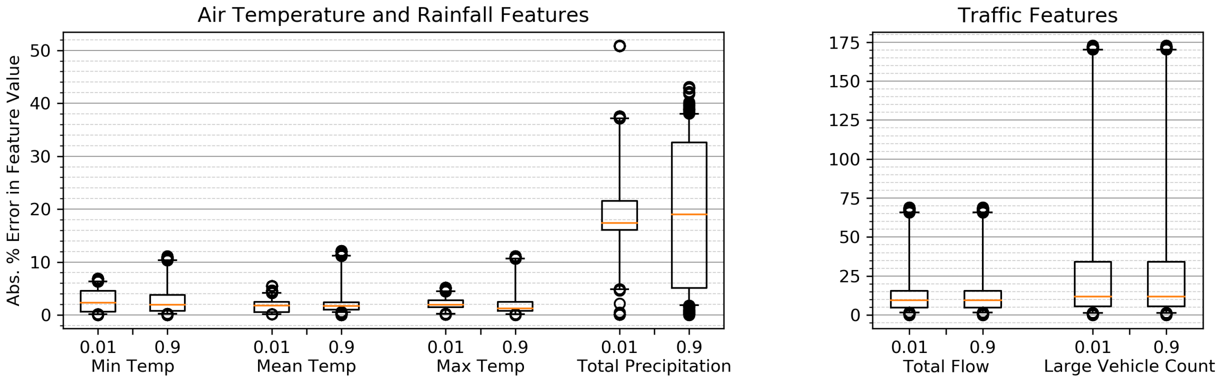


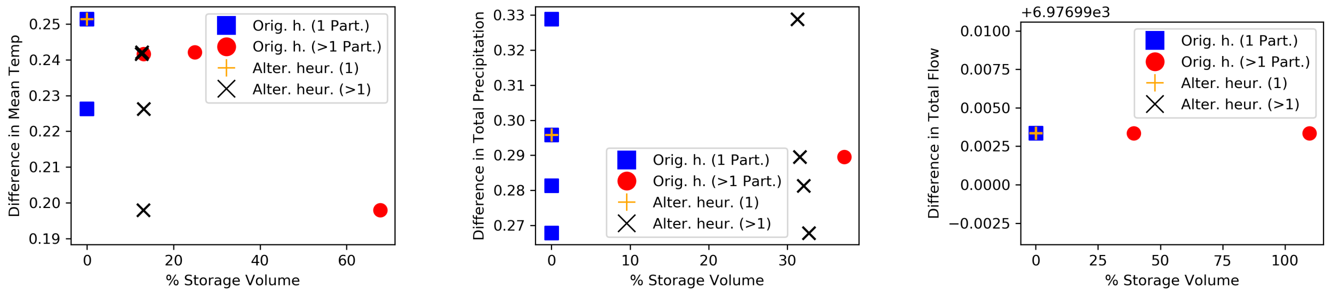
Figure 4: Error incurred by engineering features using the reduced datasets (created using the alternative heuristic function). Results are shown for $\alpha = 0.01$ and $\alpha = 0.9$.

during the linking process. Each of these are described further in this section.

First, we found that the alternative heuristic led to a lower maximum error observed in the engineered features compared to the original heuristic function, when the number of partitions output is greater than 1. Furthermore, the original heuristic was not more accurate compared to the alternative heuristic, given approximately the same storage volume used. This lack of counter-result does not prove hypothesis H1, yet our results do support the hypothesis.

Second, we found that supplementary instances that were far from the primary instances were less accurately reconstructed after being reduced with the alternative heuristic, compared to the original heuristic. This conclusion supports hypothesis H3 and no result showed the alternative heuristic yielding a more accurate reconstruction of the original supplementary features compared to the original heuristic.

Finally, no partition in the reduced dataset that did not overlap with the primary instances stored more than a single model coefficient when the alternative heuristic was used, supporting hypothesis H2. In comparison, multiple partitions that did not overlap with the area applicable to the primary dataset retained more than 1 model heuristic when the original heuristic was used. These coefficients retained information that was not useful for the linking process, giving a less efficient reduction of the supplementary

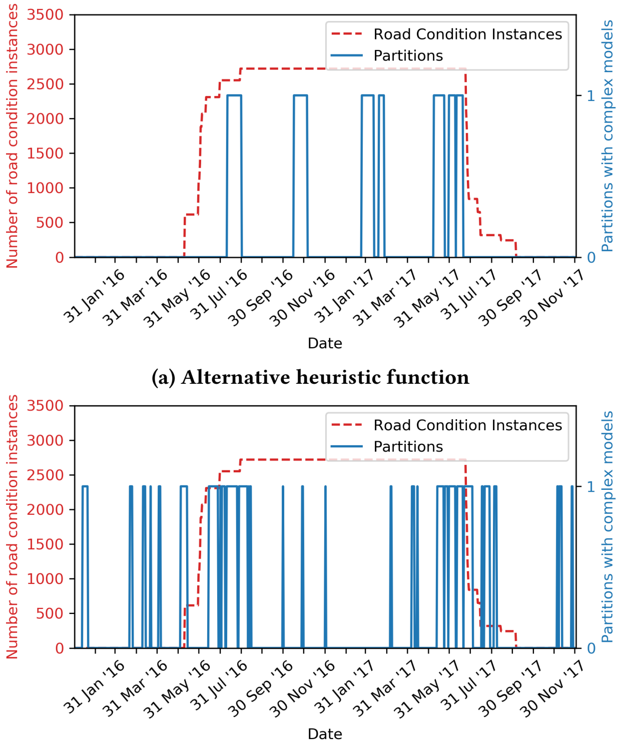


(a) Daily mean temperature

(b) Daily total precipitation

(c) Daily total traffic count

Figure 5: Error in engineered features versus storage used by the reduced datasets created using the original and alternative heuristic functions. Note, when the dataset contained 1 partition we expect the reduced datasets to achieve approximately the same results, and only expect different results when the number of partitions is greater than 1.



(a) Alternative heuristic function

(b) Original heuristic function

Figure 6: Temporal locations of partitions with more than one model coefficient for the air temperature dataset.

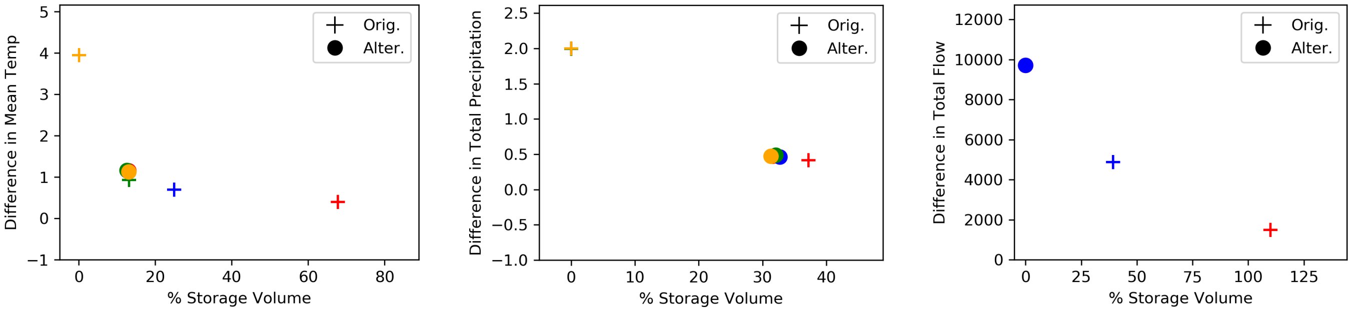
datasets. However, the variance of the instances within the partition also affected the number model coefficients stored for that partition. An unexplored question remains – to what extent was the number of coefficients stored for a partition attributed to magnitude of model improvement versus the number of primary instances that model was linked to?

Whilst these results are useful and show the utility of the alternative heuristic function, in this paper we only considered cases where the spatial and temporal distributions of the primary dataset are known prior to reduction. In several scenarios, this distribution can be inferred from previous samples of data. For example, weather and traffic data, such as the datasets used for evaluation in this paper, are stored in monthly and yearly datasets. Since the distribution of road sections does not change significantly from year to year, we may use the spatio-temporal distribution of a road network from a previous year during the reduction of the latest weather and traffic data. However, in cases where the spatio-temporal distribution may not be known, it may be beneficial to retain information in areas and time periods near to the area/time period applicable to the primary dataset, as well as those directly applicable. Furthermore, since the alternative heuristic caused a significant slowdown in the reduction process, estimating the feature engineering error without requiring the linking procedure to be completed would allow for a faster reduction process.

7 CONCLUSION

The growth of urban data has led to datasets that are too large to be processed on a single machine, yet they contain high redundancy and significant autocorrelation. In common dataset linking scenarios, the quantity of data present can make the linking process slow or infeasible. Whilst the kD-STR algorithm [17, 18] has been shown to be effective at reducing the quantity of data in a spatio-temporal dataset, it does not consider the properties of a primary dataset when reducing a supplementary dataset. This can lead to a less efficient reduction that minimises information loss in areas and time periods not applicable to the primary dataset whilst incurring unnecessary error in those that are applicable.

In this paper we presented a preprocessing step and alternative heuristic function for kD-STR that overcomes these issues. The preprocessing step removes unnecessary supplementary information and the alternative heuristic, unlike the original kD-STR heuristic, considers the error incurred in the feature engineering stage of the data linking process. The alternative heuristic was demonstrated to give a reduced dataset that was more accurate when using the reduced dataset to augment a primary dataset, whilst using comparable or lower storage volumes than the original kD-STR heuristic. However, this was shown to give a less accurate reconstruction of



(a) Daily mean temperature

(b) Daily total precipitation

(c) Daily total traffic count

Figure 7: Error in original supplementary dataset features versus storage used by the reduced datasets created using the original and alternative heuristic functions. Results are shown only for those cases where the reduced datasets contained more than 1 partition.

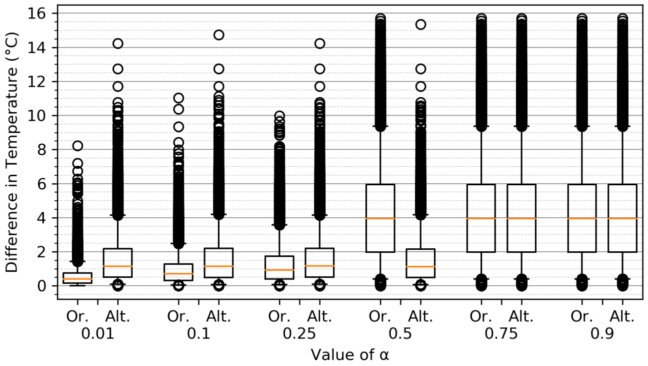


Figure 8: Error in mean temperature feature for the original supplementary dataset after being reconstructed from the reduced air temperature dataset. Results are shown for the alternative and original heuristic functions.

the original dataset in those partitions not applicable to the primary dataset.

These results improve the error incurred by kD-STR which considerably reduces the time taken to link spatio-temporal datasets. Such speedups improve the efficiency of data analysis in both research and industry, and allow greater quantities of data to be analysed. Future work in this area could focus on improving the alternative heuristic for scenarios where the primary dataset's spatial and temporal distributions are unknown or estimated. Adaptations of kD-STR for scenarios other than dataset linking could also be considered, as well as ways of making the algorithm distributed with the aim of increasing the size of the dataset that can be reduced.

ACKNOWLEDGMENTS

The lead author gratefully acknowledges funding by the UK Engineering and Physical Sciences Research Council (grant no. EP/L016400/1), the EPSRC Centre for Doctoral Training in Urban Science, and funding from TRL.

REFERENCES

[1] 2006. Met Office. MIDAS: UK Daily Rainfall Data. NCAS British Atmospheric Data Centre. https://catalogue.ceda.ac.uk/uuid/954d743d1c07d1dd034c131935db54e0. Accessed 01/09/2019.

[2] 2006. Met Office. MIDAS: UK Daily Weather Observation Data. NCAS British Atmospheric Data Centre. https://catalogue.ceda.ac.uk/uuid/954d743d1c07d1dd034c131935db54e0. Accessed 01/09/2019.

[3] 2019. Highways England Network Journey Time and Traffic Flow Data. https://webtris.highwaysengland.co.uk. Accessed 01/09/2019.

[4] 2019. Highways England Pavement Management System Road Condition Survey. Closed dataset, accessed 01/09/2019.

[5] Mohammad Akbari, Farhad Samadzadegan, and Robert Weibel. 2015. A Generic Regional Spatio-Temporal Co-Occurrence Pattern Mining Model: a Case Study for Air Pollution. *Journal of Geographical Systems* 17, 3 (01 Jul 2015), 249–274. https://doi.org/10.1007/s10109-015-0216-4

[6] Louai Alarabi, Mohamed F. Mokbel, and Mashaal Musleh. 2018. ST-Hadoop: A MapReduce Framework for Spatio-Temporal Data. *Geoinformatica* 22, 4 (Oct. 2018), 785–813. https://doi.org/10.1007/s10707-018-0325-6

[7] Michael Chipeta, Dianne Terlouw, Kamija Phiri, and Peter Diggle. 2017. Inhibitory Geostatistical Designs for Spatial Prediction Taking Account of Uncertain Covariance Structure. *Environmetrics* 28, 1 (2017), e2425. https://doi.org/10.1002/env.2425 e2425 env.2425.

[8] Eric M. Delmelle. 2014. *Spatial Sampling*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1385–1399. https://doi.org/10.1007/978-3-642-23430-9_73

[9] Ye Ding, Yanhua Li, Ke Deng, Haoyu Tan, Mingxuan Yuan, and Lionel M. Ni. 2017. Detecting and Analyzing Urban Regions with High Impact of Weather Change on Transport. *IEEE Transactions on Big Data* 3, 2 (2017), 126–139.

[10] Christopher R. Knittel, Douglas L. Miller, and Nicholas J. Sanders. 2016. Caution, Drivers! Children Present: Traffic, Pollution, and Infant Health. *The Review of Economics and Statistics* 98, 2 (2016), 350–366. https://doi.org/10.1162/REST_a_00548

[11] Xiangjie Kong, Menglin Li, Jianxin Li, Kaiqi Tian, Xiping Hu, and Feng Xia. 2019. CoPFun: An Urban Co-Occurrence Pattern Mining Scheme Based on Regional Function Discovery. *World Wide Web* 22, 3 (5 2019), 1029–1054. https://doi.org/10.1007/s11280-018-0578-x

[12] Sriram Lakshminarasimhan, Neil Shah, Stephane Ethier, Scott Klasky, Rob Latham, Rob Ross, and Nagiza F Samatova. 2011. Compressing the Incompressible with ISABELA: In-situ Reduction of Spatio-temporal Data. In *Euro-Par 2011 Parallel Processing*, Emmanuel Jeannot, Raymond Namyst, and Jean Roman (Eds.). Springer Berlin Heidelberg, 366–379. https://doi.org/10.1007/978-3-642-23400-2

[13] Bei Pan, Ugur Demiryurek, Farnoush Banaei-Kashani, and Cyrus Shahabi. 2010. Spatiotemporal Summarization of Traffic Data Streams. In *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming*. ACM, 4–10. https://doi.org/10.1145/1878500.1878504

[14] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572.

[15] David Sathiaraj, Thana on Punkasem, Fahui Wang, and Dan P.K. Seedah. 2018. Data-driven Analysis on the Effects of Extreme Weather Elements on Traffic Volume in Atlanta, GA, USA. *Computers, Environment and Urban Systems* 72 (2018), 212 – 220. https://doi.org/10.1016/j.compenvurbsys.2018.06.012

[16] Richard Sinnott and W. Voorsluys. 2016. A Scalable Cloud-Based System for Data-Intensive Spatial Analysis. *International Journal on Software Tools for Technology*

Transfer 18, 6 (01 Nov 2016), 587–605. https://doi.org/10.1007/s10009-015-0398-6

[17] Liam Steadman, Nathan Griffiths, Stephen Jarvis, Mark Bell, Shaun Helman, and Caroline Wallbank. 2020. kD-STR: A Method for Spatio-Temporal Data Reduction and Modelling.

[18] Liam Steadman, Nathan E Griffiths, Stephen A Jarvis, Stuart McRobbie, and Caroline Wallbank. 2019. 2D-STR: Reducing Spatio-temporal Traffic Datasets by Partitioning and Modelling.. In *GISTAM*. 41–52.

[19] Athanasios Theofilatos. 2017. Incorporating Real-Time Traffic and Weather Data to Explore Road Accident Likelihood and Severity in Urban Arterials. *Journal of Safety Research* 61 (2017), 9 – 21. https://doi.org/10.1016/j.jsr.2017.02.003

[20] Senzhang Wang, Xiaoming Zhang, Jianping Cao, Lifang He, Leon Stenneth, Philip S. Yu, Zhoujun Li, and Zhiqiu Huang. 2017. Computing Urban Traffic Congestions by Incorporating Sparse GPS Probe Data and Social Media Data. *ACM Trans. Inf. Syst.* 35, 4, Article 40 (July 2017), 30 pages. https://doi.org/10.1145/3057281

[21] Kesheng Wu, Dongeun Lee, Alex Sim, and Jaesik Choi. 2017. Statistical data reduction for streaming data. In *2017 New York Scientific Data Summit (NYSDS)*. 1–6. https://doi.org/10.1109/NYSDS.2017.8085035

[22] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. 2018. Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. ACM, New York, NY, USA, 984–992. https://doi.org/10.1145/3219819.3219922

[23] Da Zhang and Mansur R. Kabuka. 2017. Combining Weather Condition Data to Predict Traffic Flow: A GRU Based Deep Learning Approach. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*. 1216–1219. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.194

APPENDIX

A KD-STR ALGORITHM

The kD-STR algorithm, adapted to use the new heuristic presented in this paper, is shown in Algorithm 2. The algorithm begins by forming a partition tree over D_i. Then, beginning at the root of the tree with all instances belonging to a single partition, all data within the partition is modelled using a single model coefficient. Then, the heuristic value (Equation 1) is calculated for this reduction. Next, kD-STR iteratively decides between increasing the number of partitions in the reduced dataset and the number of model coefficients stored for each partition. The algorithm stops when the heuristic cannot be minimised further, i.e., $h_1 \geq h$ and $h_2 \geq h$, and the set of partitions and models $\langle P_i, M_i \rangle$ is output.

The kD-STR algorithm takes a supplementary dataset, sample of the primary dataset and the parameter α as input. Here, α is a weighting coefficient that balances reduction in storage with reduction in introduced error (NRMSE), and $0 \leq \alpha \leq 1$. Since a value of $\alpha = 0$ indicates a preference for minimising the error introduced with no consideration of the storage used, kD-STR would not stop iterating until the error (Equation 3) is 0. Such a perfect model may be unrealistic and less efficient than storing the original data. Conversely, when $\alpha = 1$, both increasing the complexity of a model and increasing the number of partitions would increase the storage used, thus only a single partition and model coefficient is stored for the entire dataset. Both of these scenarios may not be useful and so these values should be avoided for α.

More information about the kD-STR algorithm can be found in literature [17, 18].

Algorithm 2: The kD-STR algorithm for reducing supplementary datasets

```
Input: $D_i, D_{\text{prim}'}, \alpha$
Output: $\langle P_i, M_i \rangle$
clusterTree = cluster($D_i$);
numberClusters = 1;
$P_i$ = findPartitions($D_i$, clusterTree, numberClusters);
$M_i = \{\}$ ;   // Initialise the set of models to the empty set
for $p_j$ in $P_i$ do
    $m_j$ = model($D_{i,j}$, 1) ;   // Model the data in partition $p_j$
        from $D_i$ using the simplest complexity
    $M_i$.add($m_j$);
$h$ = heuristic($D_i, D_{\text{prim}'}, P_i, M_i$) ; // Calculate heuristic for 1
    partition and simple model
// Now iterate until heuristic $h$ is minimised
do
    // First, check if increasing an existing model's
        complexity minimises $h$ further
    $h_1 = h$;
    for $p_j$ in $P_i$ do
        $M'_i = M_i$;
        $m'_j$ = model($D_{i,j}$, $m_j$.complexity + 1);
        $M'_i$.replace($m_j, m'_j$) ;   // Replace $m_j$ with $m'_j$ in $M'_i$
        $h'$ = heuristic($D_i, D_{\text{prim}'}, P_i, M'_i$);
        if $h' < h_1$ then
            $h_1 = h'$;
            $M_{best} = M'_i$;
    // Second, check if increasing the number of partitions
        minimises $h$ further
    $P'_i$ = findPartitions($D_i$, clusterTree, numberClusters+1);
    $M''_i = \{\}$;  // Initialise the set of models to the empty set
    for $p_j$ in $P'_i$ do
        if $p_j$ in $P_i$ then
            $M''_i$.add($m_j$) ;   // Add $m_j \in M_i$ to $M''_i$
        else
            $m''_j$ = model($D_{i,j}$, 1);
            $M''_i$.add($m''_j$);
    $h_2$ = heuristic($D_i, D_{\text{prim}'}, P'_i, M''_i$);
    // Finally, if increasing the number of partitions, or the
        complexity of an existing model, is more optimal than
        $h$, take that choice
    if $h_1 < h_2$ and $h_1 < h$ then
        $M_i = M_{\text{best}}$;
        $h = h_1$;
    else if $h_2 < h_1$ and $h_2 < h$ then
        $P_i = P'_i$;
        $M_i = M''_i$;
        $h = h_2$;
        numberClusters = numberClusters + 1;
while $h_1 < h$ or $h_2 < h$;
return $P_i, M_i$
```