

Perl Status Reporter (SRr) on Spatiotemporal Data Mining

Sugam Sharma, and Shashi Gadia

Department of Computer Science, Iowa State University, USA

sugam.k.sharma@gmail.com

ABSTRACT

Perl Status Reporter (SRr) [1] is a data mining tool and in this research work we illustrate the use of SRr on spatio-temporal database. Spatio-temporal data are associated with time and space and used vastly in different applications of diversified areas such as geography, geology, city planning, agriculture, environmental study, traffic navigation, aerospace industries, and so on. We exploit spatio-temporal data which is composed of road traffic data of metro cities of United States of America. The traffic data file consists of traffic information, based on spatial location and time, in metro cities and we want to extract the piece of information from that huge file which is of our interest. SRr needs a flat data input file to perform the data extraction process. SRr reads input file and derives the desired information from it. The input file has two parameters of interest (firstKey and secondKey). The association of these two parameters describes the uniqueness of entries in that file and works as composite key similar to that in traditional databases. Based on the parameter of interest SRr generates an output file, named as firstKey_secondKey. SRr subsequently reads the input file line by line and extracts the data of interest and stores all the tuples having the same composite key, into that output file generated by SRr.

KEYWORDS

Perl, regexp, Spatio-temporal data, Traffic data.

I. INTRODUCTION

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help one to focus on the most important information in their huge data warehouses. There are many upcoming tools being developed every day for text/data mining process. Perl SRr is one more development in the same direction and for the same purpose. The beauty of SRr over existing tools is that SRr is completely written in Perl and is highly customizable. The scope of operations of SRr is not only limited to data mining in Bioinformatics but also data retrieval in other areas as well. SRr is a multipurpose tool, well efficient in information extraction process.

In this research work, we apply Perl SRr on traffic database to extract the desired traffic data for subsequent analyzation. The input file is a flat file in textual format containing traffic data. The first column entry is the identification (Id) of the city where the traffic has been collected from. This Id is unique for a particular metro city. The second column entry is the name of that metro city. The third column entry is the name of the company responsible for data collection in the respective metro city area.

Next column entry is the date, on the traffic data has been collected. Rest of the column entries are self informative and are not much important in our development. Column entries first and third are of our interest and we call them as parameters of interest. In the database file, we call the first column as firstKey and third column as secondKey and combination of both of them forms a

composite key. In this work there is a key factor which is the base for information retrieval and in our implementation firstKey and secondKey are the main factors for data fetching.

SRr reads the database, one tuple at a time, eradicates its separator, and stores the contents of that tuple in an array. The filled array is further operated to extract firstKey and secondKey. SRr joins these keys together and forms a composite key subsequently. SRr keeps tracking the traffic data and generates output files based on the composite key. The name of the file will be the same as the composite key. The extracted data consisting of the same composite key will be stored in that generated output file. Thus for an individual composite key, SRs generates a separate output file and fills it in the same fashion as stated above.

Many real world objects are associated with time and space. There are good number of applications such as GPS (Global Positioning Systems), mobile phone users within mobile networks, wireless communication network, and environmental monitoring systems etc which handle and manage spatially and temporally referenced data. In the past, research in spatial and temporal data models and database systems has mostly been done independently. Spatial database research has focused on modeling and querying geometries associated with objects while temporal databases have focused on modeling and querying temporally evolving data. Since temporal and spatial data models have been intensively researched in the temporal and the spatial database communities, one may consider approaches to combine temporal and spatial data models to build spatiotemporal data. In this research work we use the spatiotemporal traffic data which provides the information about the traffic condition in terms of time and space dimension. The traffic data is collected through different medium: 1) road side static sensors disseminate the data to static sink, 2) data supplied by data collection agencies such as Traffic.com, and Navteq.com etc. One can have the feasibility to fetch the real time up-to-date data and old data as well. This data can be seen in textual format or in graphical format. In this research work, we just focus on the textual representation of the data. The graphical representation of the extracted data is the future research work. The rest of the paper is organized as follows.

Section II discusses about the spatial, temporal and spatio-temporal data. Section III is the data management process section. Section IV is the result section. This section consists of input sample file we use in our implementation and sample output produced. Section V is the related work and paper is concluded in section VI.

II. SPATIAL, TEMPORAL, and SPATIO-TEMPORAL DATABASES

Spatial databases are exploited vastly in different applications of diversified areas such as geography, geology, city planning, agriculture, environmental study, traffic navigation, aerospace industries, and so on. In this research work we focus on spatio-temporal database especially traffic data. We describe spatial, temporal, and spatio-temporal data in detail as follows.

A. SPATIAL DATABASES

One of the most prominent examples of spatial data is geospatial objects located in a spatial frame such as the Earth's surface. Many researchers have been motivated to store and query spatial data, resulting in spatial databases. There are two communities which extensively research on spatial data the spatial database community and the GIS community. The former community has actively researched on spatial data types, data models, and query languages which support ad-hoc queries over the data models with the data types.

On the other hand, the GIS community more focuses on developing practical methodologies to selectively represent and process geospatial data. A key issue in object-based models of spatial information is the choice of a basic set of spatial data types required to model common shapes on

maps. The spatial data is based on the OGC Geometry Object Model in the simple feature specification for SQL [2]. In the model, class *Geometry* serves as the base class and many subclasses such as *Point*, *Curve (Line)* and *Surface (Polygon)* are extended from the base class [3]. Most spatial databases do not stand on their own, instead they are just an extension to relational databases.

B. TEMPORAL DATABASES

A temporal database is capable of storing evolution of data, thereby allowing users to examine complete object histories [4]. Applications of the temporal databases include financial, record-keeping, and scientific applications [5]. Temporal databases are one of active research areas in the database community and tremendous research work has been done by many researchers. There are many different types of temporal data models and they have their own merits in their specific applications. In temporal database literature, we can find three types of timestamps - instants, intervals, and temporal elements. Based on timestamps, temporal data models are termed *point-based models*, *interval-based models*, and *temporal element-based models*.

It is important to emphasize that only temporal elements are legitimate domains of objects and events in the real world. However, a temporal element cannot be represented using a fixed length because it is defined as a finite union of intervals. Therefore, intervals or instants are used to timestamp fragments of objects that are stored in multiple fixed-length tuples. Typically such timestamps are attached at tuple level rather than value level. In general, temporal data models are introduced with their query languages. Among many temporal query languages, SQL/TP [6] and SQLT [7] use point-based data models; TSQL2 [8] and IXSQL [9] use interval-based data models; and ParaSQL [10] and NRTC [11] use temporal element-based data models.

C. SPATIOTEMPORAL DATABASES

Many real world objects are associated with time and space. GPS (Global Positioning Systems), mobile phone users within mobile networks, wireless communication network, and environmental monitoring systems are some examples of applications which handle and manage spatially and temporally referenced data. Because many applications require an ability of managing spatiotemporal data, there is a growing attention to spatiotemporal databases [12]. Spatiotemporal databases have gained considerable attentions and have been researched actively over a significant period. However, there still exist very few prototypes of complete systems, and far less products that provide effective support for applications tracking changes to spatial and non-spatial (or ordinary data) over time because the design and implementation of a complete spatiotemporal database is a challenging undertaking, involving extensions to all aspects of a non-spatiotemporal architecture such as data model, query language, query optimizer, query evaluator, programming environment, storage manager, and indexes [13]. In the past, research in spatial and temporal data models and database systems has mostly been done independently. Spatial database research has focused on modeling and querying geometries associated with objects while temporal databases have focused on modeling and querying temporally evolving data. Nevertheless, many researchers have tried to combine the two areas because they are all dealing with dimensions and are closely related [14]. Since temporal and spatial data models have been intensively researched in the temporal and the spatial database communities, one may consider approaches to combine temporal and spatial data models to build spatiotemporal data ones. There are two directions to accommodate temporal and spatial data models; 1) the embedding of a temporal awareness in spatial data models and, 2) the accommodation of space into temporal data models.

III. DATA MANAGEMENT PROCESS

As SRr is a data-mining tool, so it helps a user to retrieve information from the database but does not provide insert or update like operations. In this section we discuss the process involved in data retrieval. In this section we focus on the process involved in data retrieval.

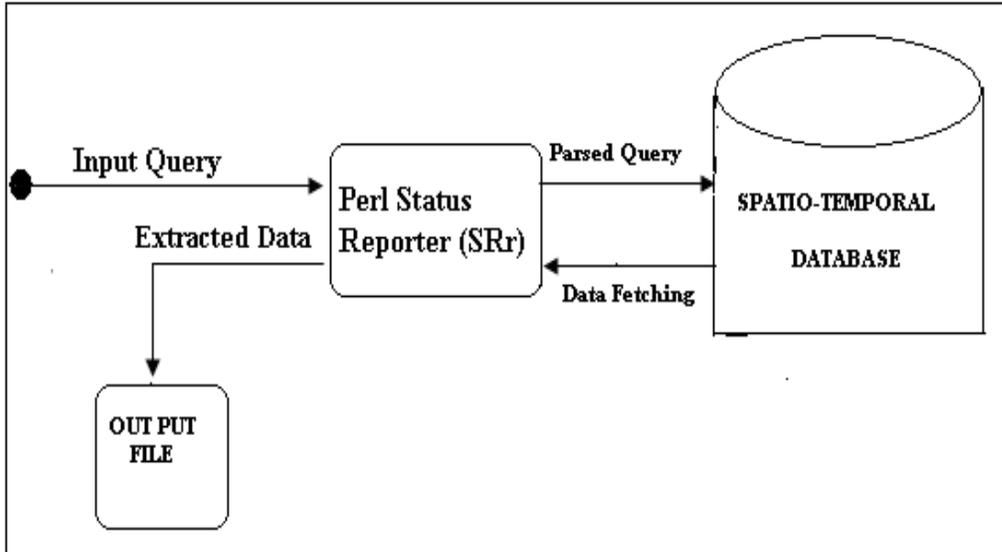


Figure 1. Process Architecture

A. PROCESS ARCHITECTURE

In figure 1, we show the complete architecture of the process involved in data retrieval. The part to the right side of SRr block, consisting of parsed query propagation, data fetching, and spatio-temporal database is the back-end. The part to the left of SRr block, consisting of input query and extracted data propagation and storage into an output file constitutes the front-end. SRr works as the middle layer and helps to establish communication between the front-end and back-end. It takes the user's query as input, and parses it. Based on the parsed query, it retrieves the desired data from the database, and stores that into an output file. In order to retrieve the desired data from a huge database, a user just needs to supply an input string, composed of the composite attributes and can see the extracted tuples of information in output file. During the entire process a user is completely unaware of the complexity involved in back-end processing.

B. ALGORITHM

In this section we write the pseudo code of the process of data retrieval from the database.

ALGORITHM 1: Data retrieval from spatio-temporal database

1. **BEGIN**
 2. User supplies an input query to SRr
 3. SRr accepts the input query and parses it
 4. SRr iterates the whole database and extracts the complete tuples of matched data
 5. SRr stores the retrieved data into an output file.
 6. **END**
-

SRr performs the process operation on flat data input file. The following code snippet describes how SRr reads the input file and parses the sentences and categories the data. We write the sample code block-wise to make easy understanding.

In the following segment we declare the variables globally. To *\$prc_file* variable we assign the input file to be processed. In our implementation we use *text_mining.txt* as the input file and it is a flat textual file.

```
my $prc_file="text_mining.txt";
my $output1_file;
my $first_key;
my $store_first_key;
my $city_Name;
my $second_key;
my $store_second_key;
```

In following block of code SRr reads a file using a file handler INPUTHANDLER. If SRr is unable to open the input file, it throws a message.

```
open INPUTHANDLER, "<$prc_file" || die "Could not open file for reading.\n";
print"\n SRr has performed the text mining operations on input file \n";
```

In the following segment SRr reads the input flat file one line at a time, splits the line based on the separator used and stores the contents in an array.

```
while (<INPUTHANDLER>) {
my@cur_line = split /,/, $_ ;
my @store_cur_line = @cur_line;
```

Now SRr extracts the data from the current line. We assign the parameters of interest in *\$first_key* and *\$second_key* variables respectively. The following block of code depicts this.

```
$first_key = shift @cur_line;
$city_Name = shift @cur_line;
$second_key = shift @cur_line;
```

The following segment is a bit confusing but not a rocket science definitely. This part of SRr just tracks the composite attributes, generates the file based on those attributes and stores those data line containing the same composite attributes in that file. SRr uses if-els statement to separate out the generation of different files. The generated files are opened in append mode and any number

of line can be added to it. SRr uses a file handler, OUTPUTHANDLER to write out the data to the file.

```

if($store_first_key eq $first_key && $store_second_key eq $second_key){

    output1_file = "$first_key"."_"."$second_key".".txt";
    open OUTPUTHANDLER, ">>$output1_file" || die "Could not open file for writing.\n";
    print OUTPUTHANDLER " @store_cur_line\n";

}

elseif($store_first_key eq $first_key && $store_second_key ne $second_key){

    $output1_file="$first_key"."_"."$second_key".".txt";
    open OUTPUTHANDLER, ">>$output1_file" || die "Couldnot open file for writing.\n";
    print OUTPUTHANDLER " @store_cur_line\n";

}

elseif($store_first_key ne $first_key && $store_second_key eq $second_key){

    $output1_file="$first_key"."_"."$second_key".".txt";
    open OUTPUTHANDLER, ">>$output1_file" || die "Could not open file for writing.\n";
    print OUTPUTHANDLER " @store_cur_line\n";

}

else {

    $output1_file="$first_key"."_"."$second_key".".txt";
    open OUTPUTHANDLER, ">>$output1_file" || die "Could not open file for writing.\n";
    print OUTPUTHANDLER " @store_cur_line\n";

}

$store_first_key = $first_key;
$store_second_key = $second_key;
}
    
```

IV. RESULTS

A. INPUT FILE SNIPPET

In this section we show the curtailed sample input data file we use in our implementation. The data in the file is road traffic data in different metro cities in United States. The first column represents an Id of the city, and second column represents the name of the city, where the traffic belongs to. The third column represents the company name which collects the traffic data for that city. Rests of the columns are informative enough to understand. Column first and third are of our interest and together they form a composite key which differentiates the line from other lines.

```

7,NewYork,TRANSCOM,11/27/2008,23:55,TRANSCOM,11254,59571,No road
selected,0,,W,THRU,5,5,,5,54,5,,5,,,
7,NewYork,TRANSCOM,11/27/2008,23:55,TRANSCOM,11255,59572,No road
selected,0,,N,THRU,5,5,,5,41,5,,5,,,
    
```

7,NewYork,TRANSCOM,11/27/2008,23:55,TRANSCOM,11256,59573,No road selected,0,,S,THRU,5,5,,5,6,5,,5,,,

7,NewYork,TRANSCOM,11/27/2008,23:55,TRANSCOM,25233,74179,No road selected,0,,E,THRU,5,5,,5,49,5,,5,,,

7,NewYork,TRANSCOM,11/27/2008,23:55,TRANSCOM,25252,74191,No road selected,0,,E,THRU,5,5,,5,45,5,,5,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31514,77443,No road selected,1,LEFT,W,THRU,5,4,14,4,77,4,1.5,4,14,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31514,77443,No road selected,2,LEFT CENTER,W,THRU,5,4,25,4,72,4,4,3,4,25,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31514,77443,No road selected,3,RIGHT CENTER,W,THRU,5,4,36,4,56,4,11,4,36,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31514,77443,No road selected,4,RIGHT,W,THRU,5,4,7,4,74,4,1,4,7,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31515,77444,No road selected,1,LEFT,E,THRU,5,4,5,4,42,4,1,4,5,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31515,77444,No road selected,2,LEFT CENTER,E,THRU,5,4,21,4,66,4,4,4,21,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31515,77444,No road selected,3,RIGHT CENTER,E,THRU,5,4,12,4,72,4,3,8,4,12,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31515,77444,No road selected,4,RIGHT,E,THRU,5,4,17,4,67,4,1,8,4,17,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31516,77445,No road selected,1,LEFT,E,THRU,5,4,7,4,75,4,.5,4,7,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31516,77445,No road selected,2,CENTER,E,THRU,5,4,20,4,52,4,7.5,4,20,,,

16,Chicago,INDOT,11/27/2008,00:00,INDOT,31516,77445,No road selected,3,RIGHT,W,THRU,5,4,20,4,54,4,4,8,4,20,,,

16,Chicago,MT,11/27/2008,00:10,MT,25040,73993,I-355,1,LEFT,N,THRU,5,5,20,5,53,5,2,6,5,14,4,2,0

16,Chicago,MT,11/27/2008,00:10,MT,25040,73993,I-355,2,CENTER,N,THRU,5,5,13,5,49,5,1,2,5,12,0,1,0

16,Chicago,MT,11/27/2008,00:10,MT,25040,73993,I-355,3,RIGHT,N,THRU,5,5,5,5,60,3,.3,5,5,0,0,0

16,Chicago,MT,11/27/2008,00:10,MT,25040,73993,I-355,4,LEFT,S,THRU,5,5,7,5,57,4,.7,5,5,2,0,0

16,Chicago,MT,11/27/2008,00:10,MT,25040,73993,I-355,5,CENTER,S,THRU,5,5,14,5,63,4,1,2,5,13,0,1,0

16,Chicago,MT,11/27/2008,00:10,MT,25040,73993,I-355,6,RIGHT,S,THRU,5,5,7,5,56,4,4,5,7,0,0,0

16,Chicago,MT,11/27/2008,00:10,MT,25041,73994,I-355,1,LEFT,N,THRU,5,5,15,5,61,5,2,5,9,4,2,0

16,Chicago,MT,11/27/2008,00:10,MT,25041,73994,I-355,2,LEFT CENTER,N,THRU,5,5,9,5,68,4,.7,5,8,0,1,0

17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18646,67685,I-795,1,,S,THRU,1,1,12,1,63,1,,0,12,,,

17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18648,67687,I-83,1,,S,THRU,1,1,28,1,61,1,,0,28,,,

17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18649,67688,RT-32,1,,E,THRU,1,1,20,1,63,1,1,1,20,,,

17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18651,67690,I-795,1,,S,THRU,1,1,42,1,70,1,1,1,42,,,

17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18652,67691,I-70,1,,E,THRU,1,1,19,1,60,1,,0,19,,,

17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18654,67693,I-97,1,,S,THRU,1,1,10,1,70,1,,0,10,,
 17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18655,67694,I-695,1,,S,THRU,1,1,108,1,66,1,1,1,108,,
 17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18658,67697,I-795,1,,S,THRU,1,1,22,1,64,1,1,1,22,,
 17,Baltimore,MDSHA,11/27/2008,00:00,MDSHA,18659,67698,I-83,1,,N,THRU,1,1,50,1,67,1,1,1,50,,
 17,Baltimore,MT,11/27/2008,00:00,MDSHA,18646,67685,I-795,1,,S,THRU,1,1,12,1,63,1,,0,12,,
 17,Baltimore,MT,11/27/2008,00:00,MDSHA,18648,67687,I-83,1,,S,THRU,1,1,28,1,61,1,,0,28,,
 17,Baltimore,MT,11/27/2008,00:00,MDSHA,18649,67688,RT-32,1,,E,THRU,1,1,20,1,63,1,1,1,20,,
 17,Baltimore,MT,11/27/2008,00:00,MDSHA,18651,67690,I-795,1,,S,THRU,1,1,42,1,70,1,1,1,42,,
 17,Baltimore,MT,11/27/2008,00:00,MDSHA,18652,67691,I-70,1,,E,THRU,1,1,19,1,60,1,,0,19,,
 17,Baltimore,MT,11/27/2008,00:00,MDSHA,18654,67693,I-97,1,,S,THRU,1,1,10,1,70,1,,0,10,,
 17,Baltimore,MT,11/27/2008,00:00,MDSHA,18655,67694,I-695,1,,S,THRU,1,1,108,1,66,1,1,1,108,,
 17,Baltimore,MT,11/27/2008,00:00,MDSHA,18658,67697,I-795,1,,S,THRU,1,1,22,1,64,1,1,1,22,,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7010,38,PA-910,1,RIGHT,E,THRU,5,5,7,5,57,4,.8,5,7,,0,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7010,38,PA-910,2,LEFT,E,THRU,5,5,3,5,41,3,.6,5,3,,0,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7010,38,PA-910,3,RIGHT,W,THRU,5,5,3,5,56,2,.4,5,3,,0,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7010,38,PA-910,4,LEFT,W,THRU,5,5,5,5,46,2,.4,5,5,,0,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7012,42,I-79,1,RIGHT,N,THRU,5,5,14,5,69,2,.8,5,12,,2,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7012,42,I-79,2,CENTER,N,THRU,5,5,44,5,58,5,2,2,5,42,,2,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7012,42,I-79,3,LEFT,N,THRU,5,5,10,5,58,5,1,5,10,,0,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7012,42,I-79,4,RIGHT,S,THRU,5,5,8,5,19,2,1,5,8,,0,
 1,Pittsburgh,MT,11/27/2008,00:00,MT,7012,42,I-79,5,CENTER,S,THRU,5,5,20,5,56,1,1,2,5,17,,3,
 1,Pittsburgh,IDOT,11/27/2008,00:00,MT,7010,38,PA-910,1,RIGHT,E,THRU,5,5,7,5,57,4,.8,5,7,,0,
 1,Pittsburgh,IDOT,11/27/2008,00:00,MT,7010,38,PA-910,2,LEFT,E,THRU,5,5,3,5,41,3,.6,5,3,,0,
 1,Pittsburgh,IDOT,11/27/2008,00:00,MT,7010,38,PA-910,3,RIGHT,W,THRU,5,5,3,5,56,2,.4,5,3,,0,

B. OUTPUT FILE SNAPSHOT

In this section we collect the results. We capture snapshots of all output files. First figure shows the progress of SRr running and rest of the snapshots show the contents segregated from the input data file.

```
C:\PERL\bin>
C:\PERL\bin>perl SRr2.pl

SRr has performed the text mining operations on input file
C:\PERL\bin>_
```

Figure 2. SRr Status Screen Display

```
1_IDOT - Notepad
File Edit Format View Help
| 1 Pittsburgh IDOT 11/27/2008 00:00 MT 7010 38 PA-910 1 RIGHT E THRU 5 5 7 5 57 4 .8 5 7 0
1 Pittsburgh IDOT 11/27/2008 00:00 MT 7010 38 PA-910 2 LEFT E THRU 5 5 3 5 41 3 .6 5 3 0
1 Pittsburgh IDOT 11/27/2008 00:00 MT 7010 38 PA-910 3 RIGHT W THRU 5 5 3 5 56 2 .4 5 3 0
```

Figure 3. Extracted data stored in generated file 1_IDOT

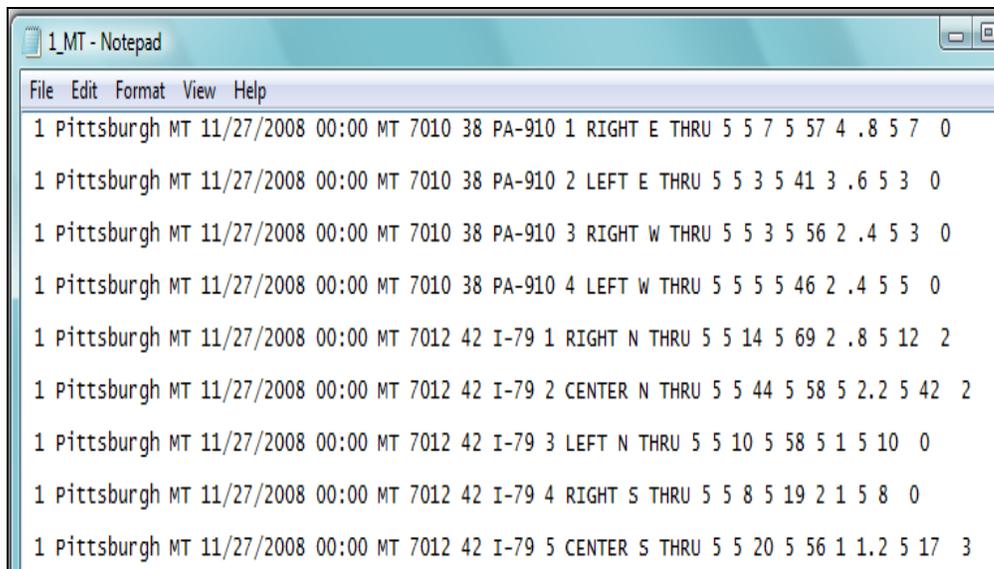


Figure 4. Extracted data stored in generated file 1_MT

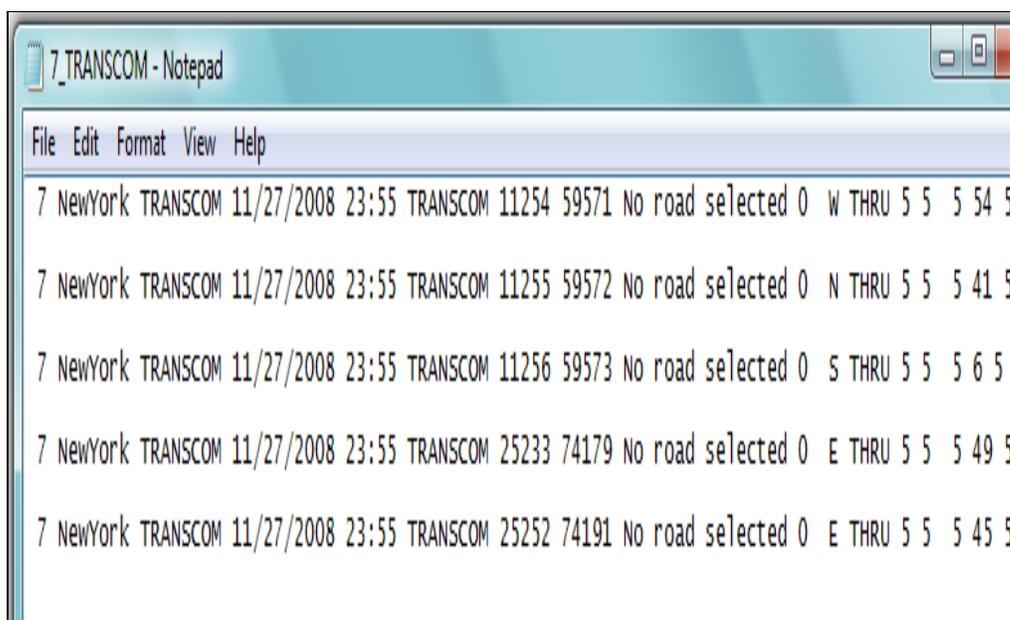


Figure 5. Extracted data stored in generated file 7_TRANSKOM

```

16 Chicago INDOT 11/27/2008 00:00 INDOT 31514 77443 No road selected 1 LEFT W THRU 5 4 14 4 77 4
16 Chicago INDOT 11/27/2008 00:00 INDOT 31514 77443 No road selected 2 LEFT CENTER W THRU 5 4 25
16 Chicago INDOT 11/27/2008 00:00 INDOT 31514 77443 No road selected 3 RIGHT CENTER W THRU 5 4 3
16 Chicago INDOT 11/27/2008 00:00 INDOT 31514 77443 No road selected 4 RIGHT W THRU 5 4 7 4 74 4
16 Chicago INDOT 11/27/2008 00:00 INDOT 31515 77444 No road selected 1 LEFT E THRU 5 4 5 4 42 4
16 Chicago INDOT 11/27/2008 00:00 INDOT 31515 77444 No road selected 2 LEFT CENTER E THRU 5 4 21
16 Chicago INDOT 11/27/2008 00:00 INDOT 31515 77444 No road selected 3 RIGHT CENTER E THRU 5 4 1
16 Chicago INDOT 11/27/2008 00:00 INDOT 31515 77444 No road selected 4 RIGHT E THRU 5 4 17 4 67
16 Chicago INDOT 11/27/2008 00:00 INDOT 31516 77445 No road selected 1 LEFT E THRU 5 4 7 4 75 4
16 Chicago INDOT 11/27/2008 00:00 INDOT 31516 77445 No road selected 2 CENTER E THRU 5 4 20 4 52
16 Chicago INDOT 11/27/2008 00:00 INDOT 31516 77445 No road selected 3 RIGHT W THRU 5 4 20 4 54
    
```

Figure 6. Extracted data stored in generated file 16_INDOT

```

16 Chicago MT 11/27/2008 00:10 MT 25040 73993 I-355 1 LEFT N THRU 5 5 20 5 53 5 2.6 5 14 4 2 0
16 Chicago MT 11/27/2008 00:10 MT 25040 73993 I-355 2 CENTER N THRU 5 5 13 5 49 5 1.2 5 12 0 1 0
16 Chicago MT 11/27/2008 00:10 MT 25040 73993 I-355 3 RIGHT N THRU 5 5 5 5 60 3 .3 5 5 0 0 0
16 Chicago MT 11/27/2008 00:10 MT 25040 73993 I-355 4 LEFT S THRU 5 5 7 5 57 4 .7 5 5 2 0 0
16 Chicago MT 11/27/2008 00:10 MT 25040 73993 I-355 5 CENTER S THRU 5 5 14 5 63 4 1.2 5 13 0 1 0
16 Chicago MT 11/27/2008 00:10 MT 25040 73993 I-355 6 RIGHT S THRU 5 5 7 5 56 4 .4 5 7 0 0 0
16 Chicago MT 11/27/2008 00:10 MT 25041 73994 I-355 1 LEFT N THRU 5 5 15 5 61 5 2 5 9 4 2 0
16 Chicago MT 11/27/2008 00:10 MT 25041 73994 I-355 2 LEFT CENTER N THRU 5 5 9 5 68 4 .7 5 8 0 1
    
```

Figure 7. Extracted data stored in generated file 16_MT

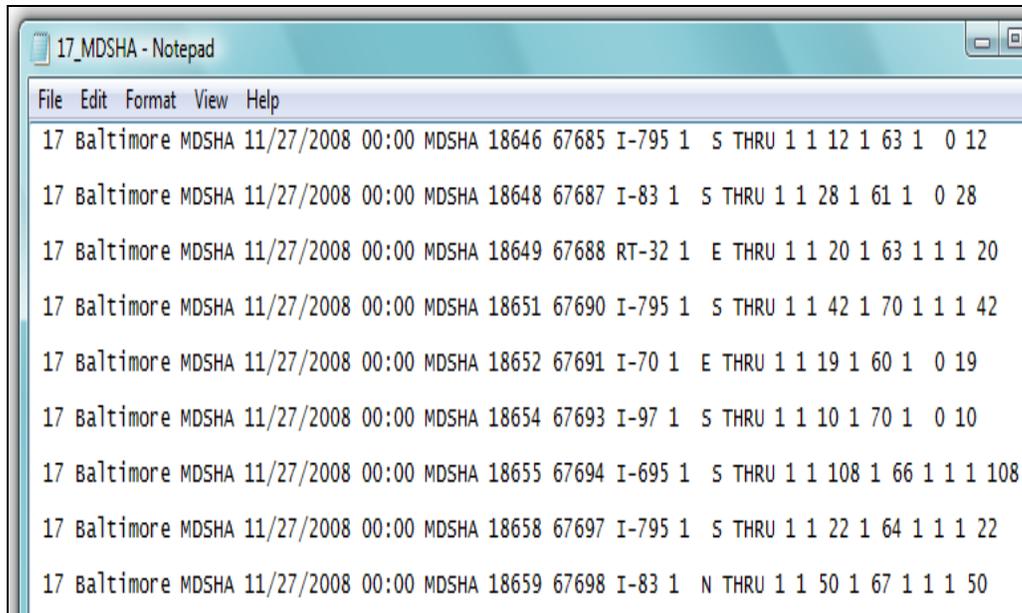


Figure 8. Extracted data stored in generated file 17_MDSHA

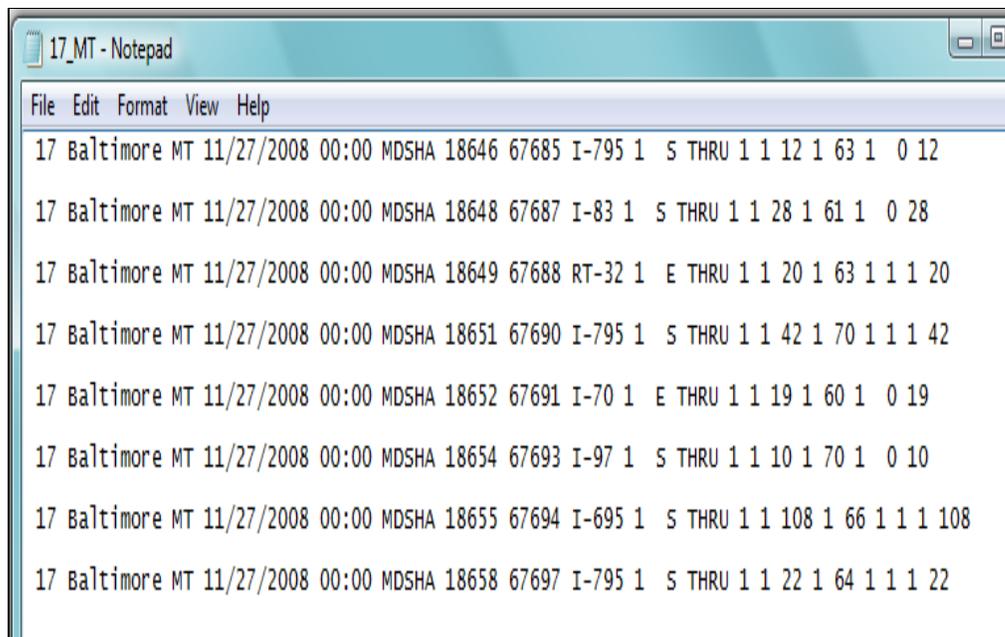


Figure 9. Extracted data stored by in generated file 17_MT

Figure 2 is a screenshot depicting the progress of SRr. Figure 3 shows the contents of the file generated by SRr. The name of the file is 1_IDOT. SRr extracts that data from the input file which has 1 in first column and IDOT in third column and generates a file with name 1_IDOT before storing the extracted data. Figure 4 shows the stored data in the file 1_MT, generated by SRr. Figure 5 shows the contents of the file 7_TRANSCOM. SRr generates a file named as 16_INDOT. Figure 6 shows the contents of that file. Figure 7 shows the generated file 16_MT and its contents. Figures 8 represents file 17_MDSHA and the data stored in that. Figure 9 is 17_MT file showing its contents extracted from the input file and stored by SRr.

V. RELATED WORK

In this section we compile the work related to spatiotemporal data. In [15] North Central Regional Association of State Agricultural Experiment Station Directors (NCRA) develops a strategy for data collection that helps in study, efficient crop management and to reduce the risk factors associated with agriculture practices, which depends on various factors. The collection and organization of data based on the states in the North Central Region, USA is mentioned as the primary goal of the Association and climate, crop and soil data is collected for these states. The result of data collection and organization efforts is the *NC94* dataset which we use extensively in our research work. In [16] authors briefly describe the attempt to calculate the degree of coldness using *NC94* spatiotemporal data in Iowa and in [17] the authors liberally explore the calculation and analysis of coldness in Iowa. It is well known fact that XML is powerful to provide configuration. In [18] the potential of XML in spatiotemporal data has been explored one again at configuration level. By varying the values of range in XML configuration file, the shades of color on map can vary distinctly which is useful to the people with defected eyesight.

VI. CONCLUSION

SRr is a data mining tool and performs mining operations on input data file and segregates the information patterns based on composite attributes. In this research work we applied SRr on spatio-temporal database consisting of traffic data of metro cities in United States. We analyzed the results generated by SRr and found that each output file has distinct data stored in it based on its composite attributes. SRr is a customizable tool and can be applied to other applications as well where one reads an input data file. It is a rare work, we have done that Perl developed mining tool, SRr extracts the spatio-temporal traffic data and segregates that for easy analyzing. We believe that our work will help computer and geographic communities, academia and industry.

ACKNOWLEDGEMENTS

Authors are thankful to Xinyuan Zhao, a Ph.D. student in Computer Science at Iowa State University (xinyuan@iastate.edu) for his helpful comments to improve this paper.

REFERENCE

- [1] S. Sharma, H. Cohly, and T. Pei, "On Generation of Firewall Log Status Reporter (SRr) Using Perl," International Journal of Network Security & Its Applications, Vol. 1 No. 2, pp. 90-99, July 2009.
- [2] Open GIS Consortium. OpenGIS Simple Feature Specification for SQL. Open GIS Consortium, 1.1 edition, 1999.
- [3] S. Shekhar and S. Shawla, editors, "Spatial Databases: A Tour," Prentice Hall, 2003.
- [4] G. Bhargava, "A 2-dimensional temporal relational database model for querying errors and updates, and for achieving zero information-loss," PhD thesis, Department of Computer Science, Iowa State University, Ames, Iowa, 1989.
- [5] C. S. Jensen and R. T. Snodgrass, "Temporal database management," IEEE Transactions on Knowledge and Data Engineering, 11(1):464-497, 1999.
- [6] D. Toman, "Point-based temporal extension of temporal SQL," In Proceedings of the 5th International Conference on Deductive and Object-Oriented Databases, pages 103-121, 1997.

- [7] C. X. Chen, J. Kong, and C. Zaniolo, "Design and implementation of a temporal extension of SQL," In Proceedings of the 19th International Conference on Data Engineering, pages 689-691, Bangalore, India, 2003.
- [8] R. T. Snodgrass, editor, "The TSQL2 Temporal Query Language," Kluwer, 1995.
- [9] N. A. Lorentzos and Y. G. Mitsopoulos, "SQL extension for interval data," IEEE Transactions Knowledge on Data Engineering, 9(3):480-499, 1997.
- [10] S. K. Gadia and S. S. Nair, "Temporal databases: A prelude to parametric data," In Temporal Databases: Theory, Design, and Implementation, pages 28-66. Benjamin/ Cummings, 1993.
- [11] A. U. Tansel and E. Tin, "The expressive power of temporal relational query languages," IEEE Transactions on Knowledge and Data Engineering, 9(1):120-134, 1997.
- [12] T. Abraham and J. F. Roddick, "Survey of spatio-temporal databases," GeoInformatica, 3(1):61-99, 1999.
- [13] T. Griffiths, A. A. A. Fernandes, N. Djafri, and N. W. Paton, "A query calculus for spatio-temporal object databases," In Proceeding of the 8th International Symposium on Temporal Representation and Reasoning, pages 101-110, 2001.
- [14] M. Erwig, R. H. Götting, M. Schneider, and M. Vazirgiannis, "Spatio-temporal data types: An approach to modeling and querying moving objects in databases," GeoInformatica, 3(3):269-296, 1999.
- [15] North Central Regional Association of State Agricultural Experiment Station Directors. Expected Outcomes. NC094: Impact of Climate and Soils on Crop Selection and Management. September 2004.
- [16] S. Sharma S., and S. K. Gadia, "On the Calculation of Coldness in Iowa, a North Central Region, United States: A Summary on XML Based Scheme in Spatial Database (NC94)," *International Conference on Advances in Information and Communication Technologies (ICT 2010)*, Springer CCIS series.
- [17] S. Sharma, and S. Gadia, "On Analyzing the Degree of Coldness in Iowa, a North Central Region, United States: An XML Exploitation in Spatial Database (NC94)," *First International Workshop on Database Management Systems (DMS 2010)*, Springer CCIS series, July 23-25, 2010.
- [18] S. Sharma, and S. K. Gadia, "An XML-Based Range Variation Approach to Render the Coldness in Iowa, a North Central Region, United States," *IEEE - Second International Conference on Information Management and Engineering (ICIME 2010)*, April 2010.

Authors

Shashi K. Gadia obtained his B.S.(Hons) and M.S. in Mathematics from Birla Institute of Technology and Science, Pilani, India, and Ph.D. in Mathematics from University of Illinois, Urbana in 1977.

His main research interest is in concept of non-atomic values in databases – values with dimensions such as time, space, and beliefs on one hand and hierarchical values such as those in XML on the other. He has worked in temporal, spatial, and multilevel security databases, optimization, incomplete information, query languages, user interfaces, pattern matching, implementation, relational, object oriented databases, and semi structured databases. He pioneered temporal databases and has been a faculty in Computer Science Department at Iowa State University, since 1986.



Sugam Sharma earned his BS (Computer Science & Engineering) from Roorkee, INDIA, MS in Computer Science from Jackson State, Mississippi, USA and currently perusing PhD in Computer Science at Iowa State University, USA.

He works in interdisciplinary research domains with following research interests: Databases-Spatial databases: Climatic data, GIS and XML, Wireless Sensor Network, Smart Home Technology, and Bioinformatics. He has worked as an Asst. professor, a researcher at Trent Lott Center for Geospatial Research, USA, and a software developer (J2EE) in NATVTEQ/TRAFFIC.COM in PA, USA. He is editor in-chief (executive), for IJCIB.

