

# Discovering Teleconnected Flow Anomalies: A Relationship Analysis of Dynamic neighborhoods (RAD) Approach

James M. Kang<sup>1</sup>, Shashi Shekhar<sup>1</sup>,  
Michael Henjum<sup>2</sup>, Paige J. Novak<sup>2</sup>, and William A. Arnold<sup>2</sup>

<sup>1</sup> Department of Computer Science, University of Minnesota, MN, USA  
{jkang, shekhar}@cs.umn.edu

<sup>2</sup> Department of Civil Engineering, University of Minnesota, MN, USA  
{henj0016, novak010, arno1032}@umn.edu

**Abstract.** Given a collection of sensors monitoring a flow network, the problem of discovering teleconnected flow anomalies aims to identify strongly connected pairs of events (e.g., introduction of a contaminant and its removal from a river). The ability to mine teleconnected flow anomalies is important for applications related to environmental science, video surveillance, and transportation systems. However, this problem is computationally hard because of the large number of time instants of measurement, sensors, and locations. This paper characterizes the computational structure in terms of three critical tasks, (1) detection of flow anomaly events, (2) identification of candidate pairs of events, and (3) evaluation of candidate pairs for possible teleconnection. The first task was addressed in our recent work. In this paper, we propose a RAD (Relationship Analysis of spatio-temporal Dynamic neighborhoods) approach for steps 2 and 3 to discover teleconnected flow anomalies. Computational overhead is brought down significantly by utilizing our proposed spatio-temporal dynamic neighborhood model as an index and a pruning strategy. We prove correctness and completeness for the proposed approaches. We also experimentally show the efficacy of our proposed methods using both synthetic and real datasets.

## 1 Introduction

This section first presents the application domain, followed by the problem statement, challenges, related work, contributions, and the scope and outline of this paper.

**Application Domain:** A teleconnection represents a strong interaction between paired events that are spatially distant from each other. A well-known example of teleconnected event pair involves the warming of the eastern pacific region (i.e. El Niño) and unusual weather patterns throughout the world [1]. In the United States, teleconnections often occur in air travel when a local weather disruption of a single airport (e.g., Chicago) causes other major airports (e.g., New York City, Atlanta, etc.) to delay or cancel flights. Indeed, many events in everyday life display patterns related to other events occurring a distance away. One type of teleconnected event of special interest to scientists occur in environmental systems when a contaminant enters a river (e.g., an oil spill) and then vanishes (e.g., the removal of the oil via natural or man-made)



**Fig. 1.** Dead Zone, Gulf of Mexico [5] (Best Viewed in Color)

downstream. Identifying these teleconnections in environmental systems is important to maintain high water quality, one of the major global challenges facing humanity according to the United Nations [2]. When contaminants enter river networks, they create problems for drinking water sources and point to the need to identify when and where the contaminant entered and exited the river network [3, 4].

For the past several years, environmental engineers and scientists have been actively studying contaminants in water by placing advanced sensors along streams or rivers [6]. One of the greatest challenges in this field, however, is to understand how contaminants *emerge* (i.e., when and where a contaminant may enter) and how they *vanish* (i.e., when and where a contaminant is removed). Pairs of *emerging* and *vanishing* events may be teleconnected. A single contaminant may *emerge* as a result of rain fall and then *vanish* downstream in natural catchments (e.g., Dead Zone in the Gulf of Mexico in Figure 1). For example, nitrate (a component of fertilizer) may emerge from storm water runoff (i.e., process of nitrification) and vanish downstream as a result of biological transformation (i.e., process of denitrification). Although there exist several known locations for vanishing events, studies using mass-balance methods show that only a fraction of the entering contaminants are “caught” [7]. Determining when and where all of these contaminants *vanish* in the river is an open area of study in environmental science with many potential benefits. For example, such research is possible to reduce economic costs and the environmental impact of contamination by limiting the location of man-made remedies [7] to areas where natural processes are shown to be inadequate for removing contamination. Thus several environmental scientists (e.g. our collaborators Novak and Arnold) have expressed the need for an efficient and robust method to discover teleconnections between these *emerging* and *vanishing* events.

There are other important and interesting applications for the discovery of teleconnected events outside the realm of environmental science as well. In transportation systems, identifying the time and the location of teleconnected congestion may be important for commuters when choosing the best route to take. In video surveillance, authorities want to be able to determine the time and source of unusual events such as unattended bags being left (i.e, *emerge*) or picked up (i.e., *vanishing*) at an airport terminal. Monitoring thousands of surveillance video streams may result in expensive manual investigations to identify these events. Thus, there is a need to efficiently detect these teleconnected relationships.

**Problem Statement:** Given a collection of sensors where each sensor has a time series of measured variables, the teleconnected flow anomaly discovery problem identifies strongly connected pairs of events. We are mostly interested in flow anomaly events and pairs of *emerging* and *vanishing* events. We define this notion informally here and formally in Section 2. Flow anomalies represent time-periods with a (user-defined) high fraction of time-instants having significantly different readings across pairs of adjacent sensors. For example, if no pollution events exist within a river, then all the observations seen at each sensor along the river will be similar. If a pollution event occurs between a pair of sensors at a single time instant, then a transient flow anomaly has been found. A persistent flow anomaly may consist of several transient flow anomalies and several observations that appears to be normal. A persistent flow anomaly found between sensors is considered dominant if it is not a subset of any other flow anomaly event occurring at this location. An *emerging* flow anomaly may be found upstream (e.g., at an industrial outfall) whereas a *vanishing* flow anomaly event may be found downstream (e.g., as a result of degradation).

Mining teleconnected flow anomaly events is computationally challenging for many reasons. First, a single flow anomaly event may consist of subsets that may not be anomalous, but are important for the event itself. This makes it difficult to use the dynamic programming principle for designing an algorithm. Second, the temporal length of each flow anomaly may vary. This makes fixed window-based paradigms unnatural. Third, there may be a large number of possible locations for *emerging* and *vanishing* flow anomaly events across all node paths and time paths in the network and all paths and time-instant paths must be searched to identify teleconnected relationships. In addition, teleconnected flow anomalies may consist of one-to-one, one-to-many, or many-to-many relationships between *emerging* and *vanishing* flow anomaly events. Identifying the relationships between flow anomaly events creates a large number of combinations across the entire network. Finally, the length of time series may be very large due to the potentially infinite nature of time.

**Related Work:** To the best of the authors' knowledge, no techniques have been reported in the literature to find flow anomalies across an entire network and then identify the relationship between these events. The most related technique, called SWEET, is our preliminary work [8] that introduced the problem of discovering flow anomalies for a pair of adjacent sensors addressing the first critical task identified in the abstract for the overall problem of discovering teleconnected flow anomalies. Computation time for SWEET was reduced significantly by introducing the concepts of a smart counter and a pruning strategy. Briefly, the smart counter allowed SWEET to scan the time series once to identify the transient flow anomalies and the pruning strategy reduced the number of candidates (i.e., time periods) to be analyzed. These algorithmic innovations reduced computation time costs by orders of magnitude. For example, for a long time series, SWEET reduced the execution time from hours to seconds. However, SWEET is limited to finding flow anomalies between only two sensors and cannot identify the teleconnected relationship between multiple flow anomalies occurring at different locations and time periods.

In order to make this paper complete, the related work on flow anomalies presented in our previous work [8] is also presented here. Related literature to flow anomalies may appear to occur in string matching, time series analysis, data stream correlations, clustering, and outlier detection. In string matching, Amir et al. uses an inverse string matching method that maximizes and minimizes the number of mismatches [9], and Lee et al. proposes a similar method using wild cards [10]. However, these techniques use an exact matching technique whereas flow anomalies are found using a statistical measure because an exact match may not occur in our problem domain. In time series analysis, several methods assume that the basic property of dynamic programming of sub-optimal substructure exists in their problem domain (e.g. [11]). However, a persistent flow anomaly may have subsets that may not be anomalous which violates this basic principle of dynamic programming. In data stream correlations, relationships between streams are identified using a correlation measure and a fixed sliding window. Chan et al. found local correlations between multiple data streams using a sliding window [12]. Global relationships between data streams were also found using a sliding window to summarize the entire data stream [13]. Multiple pre-defined sliding windows were used to find correlations based on a query [14]. Rarity and similarity of data streams were found using a fixed sliding window [15]. However, use of a fixed window presupposes that the domain specialist knows the duration of the unexpected event (e.g. Rain Events). Also, there may be multiple events occurring between multiple data streams having anomalous events of variable sizes. In clustering, methods that focus on moving clusters (e.g. [16]) or cluster transitions (e.g. [17]) often require the need of spatially dense datasets to identify each cluster. However, flow anomalies may exist in spatially sparse datasets, limiting the ability of these clustering techniques to discover each event. Basic outlier detection techniques (e.g. t-test [18]) may detect transient flow anomalies and persistent flow anomalies (at 100% mismatched time instants) if flow is considered (e.g. [19, 20]). However, these techniques are limited in finding all persistent flow anomalies since they may miss several patterns when the mismatched time instants is less than 100%.

Identifying relationships across multiple sensors presents several challenges such as identifying whether a pair of flow anomaly events that may be spatially distant are in fact related based on their spatio-temporal neighborhood. Existing approaches have modeled these relationships as a spatial neighborhood using concepts such as modeling vector fields (e.g. [21]). However, teleconnected relationships cannot be found using these models to find pairs of *emerging* and *vanishing* flow anomaly events because neighborhoods are only defined by their spatial proximity. Spatio-temporal relationships have been discovered while assuming that the temporal dimension is fixed [22]. Whereas in the teleconnected flow anomaly problem, there may exist spatio-temporal relationships having variable temporal lengths.

**Contributions:** In this paper, we propose a Relationship Analysis of spatio-temporal Dynamic neighborhoods (RAD) approach for steps 2 and 3 (identified in abstract) of the overall problem of that utilizes several inherent properties of the problem to efficiently identify teleconnected flow anomaly events across an entire network. In summary, this paper makes the following contributions:

1. We define new key concepts that utilize our proposed spatio-temporal dynamic neighborhood model.
2. We propose a new interest measure to discover teleconnected flow anomalies
3. We propose a novel RAD method to discover teleconnected flow anomalies.
4. We propose several design alternatives: “On the Fly”, spatio-temporal Dynamic Neighborhood, and a pruning strategy.
5. We prove the correctness and completeness of all proposed approaches.
6. We experimentally evaluate our proposed methods using synthetic and real datasets.

**Scope and Organization:** The following issues are beyond the scope of this paper: (i) inferring the travel time from the dataset, that is, the travel time is given as part of the input for the teleconnected flow anomaly problem, (ii) sensor placement within the network (e.g. [23]), (iii) non-point source flow anomalies (1:M and M:N) are not discovered, that is, flow anomalies only occur between adjacent sensors, (iv) complex networks, that is, only a tree network is examined in this paper, (v) only singleton neighborhoods are explored, (vi) anomalies occurring beyond the set of known sensors in the network, and (vii) arbitrary event relationships, that is, only emerging and vanishing event types are explored.

The rest of the paper is organized as follows. Section 2 presents the basic concepts and the problem statement of discovering teleconnected flow anomalies. Section 3 presents our proposed RAD method, its design decisions, and theoretical analysis. Section 4 gives the experimental evaluation and Section 5 concludes the paper and discusses future work.

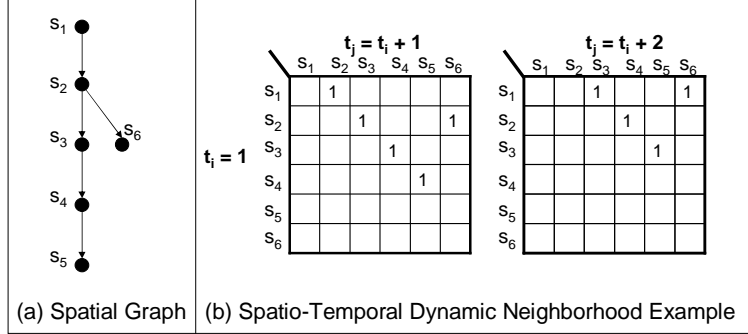
## 2 Key Concepts and Problem Statement

In this section, we first introduce key concepts for modeling the spatio-temporal dynamic neighborhood relationship and then, introduce definitions to characterize teleconnected flow anomalies. Finally, we give a formal description of the problem statement. Figure 2 illustrates the spatio-temporal dynamic neighborhood model with six spatio-temporal locations where the distance between each spatial neighbor is one unit length. Figure 3 depicts the discovery of *Emerging* and *Vanishing* Flow Anomalies respectively. In this example, the input and output is simplified for illustration by using a unit length of 1 between each sensor and assuming the travel time at each instant is given.

### 2.1 Key Concepts

A spatio-temporal set  $ST$  is denoted as  $ST = \{st_1, st_2, \dots, st_m\}$ , where  $st_i = \{s_i, t_i\}$  and  $s_i$  represents a spatial location and  $t_i$  represents a time instant. Figure 2 gives an example of six locations,  $\{s_1, s_2, \dots, s_6\}$ . A sensor observation,  $f(st_i)$ , may be associated with  $(s_i, t_i)$ .

A vector (e.g. velocity) field,  $V(st)$  or  $V(s, t)$ , is also associated with  $ST$  where  $s$  is the spatial location of the sensor and  $t$  is a time instant that maps each  $\{s_i, t_i\}$  to a velocity vector.



**Fig. 2.** Spatio-Temporal Dynamic Neighborhood Example

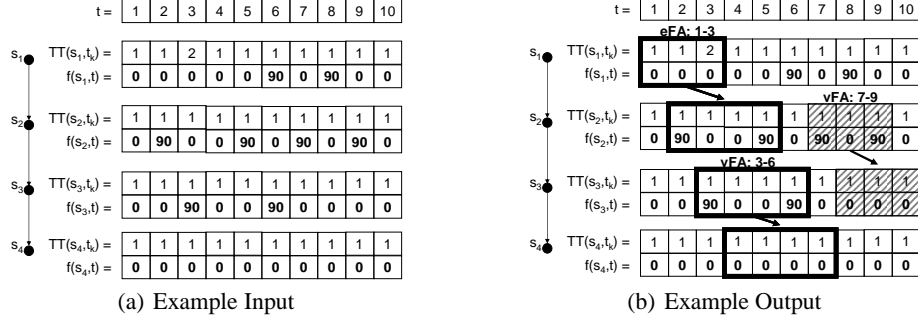
**Definition 1.** A spatio-temporal dynamic neighborhood relationship is the association between two ST locations.

Definition 1 can be formally expressed by the directed  $NB(st_i, st_j, t_k)$  where  $st_i = (s_i, t_i)$  is a neighbor of  $st_j = (s_j, t_j)$  if and only if a particle at  $s_i$  in time instant  $t_i$  will be propelled by the velocity field  $V(s, t)$  to reach location  $s_j$  on time instant  $t_j$ .

Figure 2 illustrates an example of a spatio-temporal dynamic neighborhood. For illustrative purposes only, suppose the velocity field in this example is a constant function valued 1, i.e., it is a uniformly flowing field with a unit speed downstream of 1 and the unit length between each spatially adjacent neighbor of 1. At  $t_i = 1$ , suppose we drop a particle at each spatial location and wait one second (i.e.,  $t_j = 2$ ). Based on the velocity field, the particle at each spatial location will travel one unit in length downstream and reach its adjacent neighbor. For example, at time instant  $t_i = 1$  and  $t_j = 2$ , the neighbor for  $s_1$  is  $s_2$ , i.e.,  $NB((s_1, 1), (s_2, 2))$ . Likewise, the neighbor for  $s_2$  is  $s_3$  and the neighbor of  $s_6$ ,  $s_3$  is  $s_4$ , and the neighbor of  $s_4$  is  $s_5$ . Suppose we wait an additional second ( $t_j = 3$ ) after we initially drop the particle at  $t_i = 1$  at each sensor. Then, the spatio-temporal neighborhood changes and the particle will travel an additional unit length. Thus, as shown in Figure 2b, the neighbor when  $t_i = 1$  and  $t_j = 3$  for  $s_1$  is  $s_3$  and  $s_6$ ,  $s_2$  is  $s_4$ , and  $s_3$  is  $s_5$  where the total distance traveled is 2 units in length. This simple example illustrates that a spatio-temporal neighborhood can change over time due to the flow within the network.

Neighbors  $N(st_i, t_k)$  of a ST location based on a spatio-temporal dynamic neighborhood relationship can be formally characterized as  $\{st_j | st_j \in ST, NB(st_i, st_j, t_k) = True\}$ , where  $t_k$  represents the travel time from  $s_i$  to  $s_j$ . Figure 2 gives an example of where the neighbor of  $s_1$  is  $s_2$  when the velocity starting at  $t_i = 1$  (i.e.,  $V(s, t) = 1$ ) and we wait one second ( $t_j = 2$ ).  $N(st_i, t_k)$  is considered a singleton neighborhood if it has only one element. Identifying neighborhoods for all paths and time-instant paths may be very challenging because a directed acyclic graph may merge and disperse, creating an exponential number of paths and time-instant paths due to flow.

A spatial neighborhood gives the relationship of adjacent locations  $s_i$  and  $s_j$ , whereas a spatio-temporal dynamic neighborhood gives the relationship of a pair of locations  $s_i$  and  $s_j$  at different travel times. For example, the spatial neighbors of  $s_1$  in Figure 3a is



**Fig. 3.** Discovering *Emerging* and *Vanishing* Flow Anomalies Example (Best Viewed in Color),  $TT(s_i, t_j)$  represents  $1/|V(s_i, t_j)|$ , i.e. travel time to downstream sensor at unit distance.

$s_2$ , whereas the spatio-temporal neighbor of  $s_1$  having a travel time of 2 (i.e.,  $t_j - t_i$ ) are  $s_3$  and  $s_6$ .

**Definition 2.** A *transient Flow Anomaly (tFA)* is a triple  $(st_i, t_k, \Theta_e)$  where the difference between corresponding observations (i.e., accounting for the velocity field) from a sensor and its neighboring sensors is larger than the given error threshold,  $\Theta_e$ .

Definition 2 can be formally expressed in Equation 1.

$$tFA(st_i, t_k, \Theta_e) \iff \{f(st_i) - AVG(f(st_j) | st_j \in N(st_i, t_k)) > \Theta_e\} \quad (1)$$

There are two types of transient flow anomalies, namely, *emerging* and *vanishing*. An *emerging* tFA (etFA) is defined by  $tFA(st_i, t_k) < -\Theta_e$  whereas a *vanishing* tFA (vtFA) is defined by  $tFA(st_i, t_k) > \Theta_e$ . For simplicity, Figure 3b gives examples of an *emerging* and *vanishing* tFAs for singleton neighborhoods. As can be seen, an *emerging* tFA occurs at time instant 1 between  $s_1$  and  $s_3$  having a value of -90 when the error threshold is 10 and a *vanishing* tFA occurs at time instant 3 between  $s_3$  and  $s_4$ .

**Definition 3.** A *persistent Flow Anomaly (pFA)* is a 6-tuple  $(s_i, t_k, t_s, t_e, \Theta_e, \Theta_p)$  if and only if  $(s_i, t_s, \Theta_e)$  and  $(s_i, t_e, \Theta_e)$  are transient flow anomalies, and at  $\Theta_p$  fraction of time instants  $t$  in time-interval  $[t_s, t_e]$  are associated with transient flow anomalies  $(s_i, t, t_k, \Theta_e)$ .

Definition 3 can be formally expressed in Equation 2.

$$pFA[s_i, t_k, t_s, t_e, \Theta_e, \Theta_p] \iff (tFA((s_i, t_s), t_k)) \& (tFA((s_i, t_e), t_k)) \& \left( \frac{\sum_{t=t_s}^{t_e} tFA((s_i, t), t_k)}{\text{time interval length}(t_e - t_s)} \geq \Theta_p \right) \quad (2)$$

Persistent flow anomalies are classified as either *emerging* when its tFAs are all etFAs, *vanishing* when its tFAs are all vtFAs; otherwise, they are neither. Figure 3b

gives an example of an epFA for the time interval from 1 to 3 between  $s_1$  and  $s_2$  having three etFAs and no vtFAs when the  $\Theta_p = 0.5$ .

**Definition 4.** A dominant persistent Flow Anomaly (dpFA) is a pFA that is not a subset of any other dpFA.

A dpFA may be characterized as either an *emerging* dpFA (denoted as eFA) or a *vanishing* dpFA (denoted as vFA) based on the type of its pFA. Figure 3 gives an example of an *emerging* dpFA during time instants 1 to 3 between ST locations  $s_1$  and  $s_2$ . According to the persistent flow anomaly definition, time instants 1 and 3 each satisfy the persistent threshold and its definition. However, time instants 1 and 3 are not a dpFA because they are a subset of a larger dpFA for period 1 to 3.

**Definition 5.** A teleconnected Flow Anomaly (telFA) is an eFA and a vFA pair that are related via a velocity field.

Intuitively, a telFA may represent a continuation (an eFA) cleaned up later (vFA) by a natural or man-made process. Definition 5 can be formally expressed in Equation 3.

$$\begin{aligned} telFA(eFA(s_i^1, t_k^1, t_s^1, t_e^1, \Theta_e^1, \Theta_p^1), vFA(s_i^2, t_k^2, t_s^2, t_e^2, \Theta_e^2, \Theta_p^2)) \forall (s_i^1, t_i^1), \iff \\ \exists (s_i^2, t_i^2) \text{ s.t. } \{t_i^1 \in [t_s^1, t_e^1]\} \text{ AND } \{t_i^2 \in [t_s^2, t_e^2]\} \\ \text{AND } \{NB(< s_i^1, t_i^1 >, < s_i^2, t_i^2 >)\} \end{aligned} \quad (3)$$

where  $s_i^1$  and  $s_i^2$  is the starting location in the eFA and the vFA respectively for the time period of  $t_s$  to  $t_e$ .

Figure 3b gives an example of one telFA consisting of one eFA (period 1-3, between  $s_1$  and  $s_2$ ) and one vFA (period 3-6, between  $s_3$  and  $s_4$ ). For simplicity, suppose that in this example, the unit length between each immediate neighbor is 1 and the velocity field is 1. When  $t_1 = 1$  and the travel time  $t_2 = 2$ , the neighbor of  $s_1$  is  $s_3$ . Likewise, at  $t_1 = 2$  and  $t_1 = 3$ , the neighbor of  $s_1$  is again  $s_3$ . A teleconnected flow anomaly may be statistically interpreted to identify emerging and vanishing events. Those events that do not satisfy the criteria for a emerging or a vanishing anomaly are not considered to be a telFA.

## 2.2 Problem Statement

The teleconnected flow anomaly discovery problem can be defined as follows:

**Given:** (1) A directed acyclic network consisting of ST locations; (2) A set of observations at each ST location for  $t = 1 \dots n$ , where  $n$  is the length of the time series; (3) The relevant aspects of the velocity field are represented by the travel time information from each sensor to its neighboring sensor at different start time instants; (4) An error threshold  $\Theta_e$ ; (5) A persistent threshold  $\Theta_p$ ; and (6) A spatial neighborhood (W-Matrix [24]) which maps the spatial locations to a boolean value.

**Find:** All Teleconnected Flow Anomaly relationships.

**Objective:** Minimize the computational costs.

**Constraints:** The directed acyclic network has a tree structure.

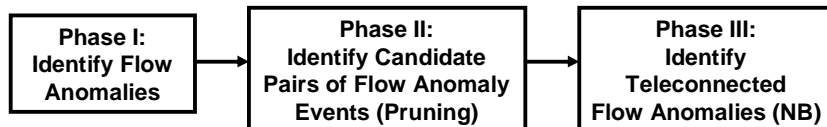


Fig. 4. RAD Approach

**Example.** Figure 3a gives an example of an input time series for four sensors where the travel time is the temporal length when one observation is expected to be made between each spatially neighborhood sensor. Figure 3b gives an example output of a teleconnected flow anomaly consisting of one eFA and one vFA when the error threshold is zero and the persistent threshold is 0.5. The eFA between ST locations  $s_1$  and  $s_2$  occurs for time period 1 to 3 and satisfies the persistent threshold, is dominant, and emerging. There are two *vanishing* flow anomalies. The first vFA occurs between  $s_2$  and  $s_3$  for the time period 7 to 9 and the second occurs between  $s_3$  and  $s_4$  for the time period of 3-6. These events are *vanishing* because the degree of change is negative. Also, they both satisfy the persistent threshold and are dominant. Based on the spatio-temporal dynamic neighborhood model, when  $t_1 = 1$  and the travel time  $t_2 = 2$ , the neighbor of  $s_1$  is  $s_3$ . Likewise, at  $t_1 = 2$  and  $t_1 = 3$ , the neighbor of  $s_1$  is again  $s_3$ . The vFA observed in period 7-9 between  $s_2$  and  $s_3$  is not linked to the eFA found between  $s_1$  and  $s_2$  because the travel time ( $t_2$ ) between  $s_1$  and  $s_2$  is not part of the neighborhood at 7-9 when the travel time  $t_1$  is between period 1-3.

### 3 Mining Teleconnected Flow Anomaly Events

In this section, we first introduce our proposed RAD (Relationship Analysis of spatio-temporal Dynamic neighborhoods) approach. We then explain key design decisions in the approach and provide its theoretical analysis.

#### 3.1 RAD Approach

This section presents the RAD (Relationship Analysis of spatio-temporal Dynamic Neighborhoods) approach to discover teleconnected flow anomalies among *emerging* and *vanishing* flow anomalies. The RAD method has three phases, namely, *identify flow anomalies*, *identify candidate pairs of flow anomaly events*, and *identify teleconnected flow anomalies* (Figure 4).

**Phase I: Identify Flow Anomalies.** This phase is concerned with identifying all the flow anomaly patterns across the entire network that satisfy the dpFA definition (Definition 4). Each pair of ST locations is analyzed based on its spatial neighborhood as defined by the W-matrix. For each pair of neighboring sensors, flow anomalies are retrieved using the SWEET<sup>1</sup> method.

<sup>1</sup> To keep this paper self-contained, key ideas of SWEET are discussed in the Related Work (Section 1). Due to space limitations, readers interested are encouraged to see [8] for details.

**Phase II: Identify Candidate Pairs of Flow Anomaly Events.** This phase is concerned with identifying pairs of *emerging* and *vanishing* flow anomalies that can be validated in the third phase. Candidate pairs are formed by the cross product of eFAs and vFAs. A **pruning strategy** is introduced to reduce the number of candidates.

**Phase III: Identify Teleconnected Flow Anomalies.** This phase is concerned with identifying all the teleconnected flow anomalies (Definition 5) based on the dpFAs found in the first phase. For a pair of *emerging* and *vanishing* flow anomalies respectively, their expected and actual travel times are found. The expected travel time is found based on the pair of time instants  $t_i$  and  $t_j$  at the time periods for the *emerging* and *vanishing* flow anomalies respectively. The actual travel time can be found “**On the Fly**” or using our proposed spatio-temporal **Dynamic Neighborhood**. If the expected travel time is the same as the actual travel time for all time instants in the *emerging* flow anomaly, then a teleconnection is found.

The composition of the phases may be executed sequentially or in a pipeline manner. A sequential approach executes Phase I until completion, followed by the second phase and then the third phase. By contrast, in the pipelined approach, Phase II is executed after a few eFAs and vFAs are determined in Phase I.

The rest of the section describes the design decisions applied in Phase II and Phase III (Due to space limitations, we omit significant design decisions made for Phase I; these are detailed in our previous work [8]). We begin with Phase III because it is easier to describe our pruning strategy for Phase II after defining our “On the Fly” and spatio-temporal Dynamic Neighborhood methods.

**On The Fly** The “On the Fly” design decision identifies the travel time between the spatio-temporal locations between an *emerging* FA and a *vanishing* FA by traversing the path between the two locations. For example, Figure 3b gives three examples of dpFAs found between ST locations: (1) Between  $s_1$  and  $s_2$  for period 1 to 3, (2) Between  $s_2$  and  $s_3$  for period 7-9, and (3) Between  $s_3$  and  $s_4$  for period 3-6. After all the dpFAs have been found, the teleconnected flow anomalies are discovered.

In this example, there is one emerging flow anomaly (eFA) between  $s_1$  and  $s_2$  and the other two are vanishing flow anomalies (vFAs). There are two possible pairs of dpFAs that may be teleconnected and will need to be analyzed. First, the eFA found between  $s_1$  and  $s_2$  and the vFA found between  $s_2$  and  $s_3$  are analyzed by checking their pairs of time instants. In the eFA, the expected travel time from time instant 1 ( $t=1$ ) and time instant 7 ( $t=7$ ) in vFA is found by taking its difference, which is 6. The actual travel time is found by traversing the path from  $s_1$  to  $s_2$ , shown in the spatial graph in Figure 3, which is a subset of the spatial graph in Figure 2a, starting at ( $t=1$ ) which is one. This on-the-fly computation may be based on a general path computation algorithm such as Dijkstra’s [25] or  $A^*$  [26], or a custom algorithm for trees. We used a custom algorithm for trees which has a linear (i.e., number of nodes and edges) complexity. For this eFA and a vFA pair, not every time instant in the eFA is a neighbor of the vFA and there is no teleconnected relationship. Then, the next eFA (between  $s_1$  and  $s_2$ ) and vFA (between  $s_3$  and  $s_4$ ) pair is evaluated. Here, a check of every time instant in the eFA with the vFA reveals a teleconnection relationship. For example, the expected travel time between the eFA and the vFA at their respective first time instant is  $3 - 1 = 2$ . Also, the travel time

---

**Algorithm 1** Generation of the Spatio-Temporal Dynamic Neighborhood (DN)

---

**Inputs:**

- The travel time at each ST location,  $TT[M][N]$

**Outputs:**

- Spatio-Temporal Dynamic Neighborhood (DN)

**Algorithm**

```
1: DN[N][N][M]  $\leftarrow$  0
2: for each pair of ST locations,  $s_i$  and  $s_j$  where  $i, j=1$  to M and a directed path exists do
3:   for each time instant,  $t_k = 1$  to N do
4:     actualTT = 0
5:     for each ST location  $s_k$  from  $s_i$  to  $s_j$  at  $t_k$  do
6:       actualTT += TT[ $s_k$ ][ $t_k$ +actualTT]
7:     end for
8:     DN[ $s_i$ ][ $s_k$ ][ $t_k$ ] = actualTT
9:   end for
10: end for
11: return DN
```

---

from  $s_1$  to  $s_3$  starting at 1 is also 2. Thus, each neighbor in eFA is a neighbor of at least one time instant of the vFA resulting in a teleconnection.

**Dynamic Neighborhood** The Dynamic Neighborhood based design decision uses a pre-computed spatio-temporal Dynamic Neighborhood to identify the actual travel times between the *emerging* and *vanishing* flow anomalies (denoted as RAD-index). Unlike the RAD-fly approach, the actual travel time can be determined using the Dynamic Neighborhood index for the RAD-index approach. If the expected and actual travel times are equal for all time instants in the *emerging* flow anomaly, then a teleconnection has been found.

Algorithm 1 gives the pseudocode for the construction of the spatio-temporal dynamic neighborhood (stDN). The stDN approach has one input consisting of the travel time (TT) required between each spatial neighbor of each node at every time instant. The travel time is generated based on the velocity field within the network. The output for Algorithm 1 is the spatio-temporal dynamic neighborhood itself.

The spatio-temporal dynamic neighborhood (DN) consists of three dimensions: (1) the starting ST location, (2) the ending ST locations that a particle may arrive at, and (3) the starting time instant. Initially, each element in the DN matrix is set to zero (Line 1 of Algorithm 1). Each pair of ST locations ( $s_i$  and  $s_j$ ) is analyzed where a directed path exists between these two locations (Line 2 of Algorithm 1). At each time instant  $t_k$  for the entire time series, the path between  $s_i$  and  $s_j$  is traversed to calculate the total travel time (Line 3-6 of Algorithm 1). The total travel time between  $s_i$  and  $s_j$  at time instant  $t_k$  can then be stored in the DN matrix (Line 8 of Algorithm 1). The process continues until all time instants are examined for each pair of ST locations and the DN is returned (Line 11 of Algorithm 1).

**Table 1.** Execution Trace for the Construction of the spatio-temporal Dynamic Neighborhood

Edge	Time Instants									
	1	2	3	4	5	6	7	8	9	10
$s_1 \rightarrow s_2$	1	1	2	1	1	1	1	1	1	1
$s_2 \rightarrow s_3$	1	1	1	1	1	1	1	1	1	1
$s_3 \rightarrow s_4$	1	1	1	1	1	1	1	1	1	1
$s_1 \rightarrow s_3$	2	2	3	2	2	2	2	2	2	-
$s_1 \rightarrow s_4$	3	3	4	3	3	3	3	3	-	-
$s_2 \rightarrow s_4$	2	2	2	2	2	2	2	2	2	-

Table 1 gives the execution trace of the construction of the spatio-temporal Dynamic Neighborhood from the example in Figure 3. The first three rows in the table give the input travel times for each edge,  $s_1$  to  $s_2$ ,  $s_2$  to  $s_3$ , and  $s_3$  to  $s_4$ . First, the pair  $s_1$  and  $s_3$  is analyzed to get the total travel times starting at  $st_1$  and arriving at  $s_3$ . The travel times are obtained at each edge from the start to its destination. For example, time instant 1, starting at  $s_1$  has a travel time of 1 to  $s_2$ . Then, at time instant 2 of  $s_2$ , the travel time is again 1. Thus, the total travel time starting at time instant 1 from  $s_1$  to  $s_3$  is 2. The travel times may vary across the times series and at multiple edges. For example, the travel time from  $s_1$  to  $s_2$  at time instant 3 is 2. The travel time from  $s_2$  to  $s_3$  at time instant 5 is 1. Thus, the travel time from  $s_1$  to  $s_3$  starting at time instant 3 has a total travel time of 3. The dashes in this table represent unknown information because the travel time is not available during part of the path. This process is continued for all node pairs and all time instant pairs until all the travel times are found as shown in Table 1.

We acknowledge that the storage cost may be an issue when the number of time instants grows, there is no periodicity, and travel time fluctuates greatly over time. We plan to address this in more detail in future work. Our current source of real data, a sensor setup at Shingle Creek, MN, does not require modeling of a large number of possibilities for travel time between adjacent sensor pair s due to periodicity, low variation in elevation, rainfall amount, and snow melt-rates.

**Pruning** A key pruning design decision can be applied to the second phase when the candidate pairs are identified. In this phase, we can prune any *vanishing* flow anomalies (vFA) where each vFA is not a neighbor of the first time instant (sTime) of an *emerging* flow anomaly. The pair of ST locations are analyzed for a single path in a tree network starting at the root node. As dpFAs are found, for any two *emerging* and *vanishing* flow anomalies, the expected travel time can be determined based on their time periods and the actual travel time can be found “On the Fly” or using the spatio-temporal dynamic neighborhood. If there exists at least one *emerging* flow anomaly whose first time instant is a neighbor to a *vanishing*, then this vFA is added to the dpFAs. All other *emerging* flow anomalies are also placed in the dpFAs. In future, we plan to explore other pruning methods such as those found on spatial relationships (e.g., ancestor-descendant) among sensors.

For example, Figure 3 gives the input and output used in this example and Table 1 contains the travel times at all node and time instant pairs. In phase 1, the first pair of

ST locations ( $s_1$  and  $s_2$ ) is analyzed for dpFAs. The SWEET technique discovers one emerging dpFA during the period of 1 to 3. The second pair of ST locations ( $s_2$  and  $s_3$ ) is analyzed and a vanishing dpFA is discovered. In the second phase, this vFA is checked for a teleconnection with the first time instant of any eFA. Examining the eFA found previously within this vFA reveals that the total travel time from  $s_1$  to  $s_2$  starting at time instant 1 is 1 as also shown in Table 1. This vFA cannot be a valid telFA with any eFAs found so far and nor can any other eFA found in the dataset be linked to this vFA. Thus, this vFA is not added as a dpFA. Finally, the last vanishing dpFA found from period 3-6 is analyzed. If we examine this vFA with the original eFA discovered earlier we find that the total travel time from  $s_1$  to  $s_3$  at time instant 1 is 3 and that the expected travel time between the eFA and vFA pair is also 3 at the first time instant. Thus, this vFA is a possible telFA and is considered for evaluation.

**Lemma 1.** *The pruning based on the first time instants is a true filter, i.e., it does not eliminate any teleconnected flow anomalies.*

*Proof.* In the second phase of RAD, all dpFAs are initially found using SWEET [27] and then the vanishing flow anomalies are pruned if the first time instant of an emerging flow anomaly is not its neighbor and violates the telFA definition (Definition 5). Thus, no telFA patterns will be missed in the second phase for both approaches. In the third phase of RAD, only the pairs of emerging and vanishing flow anomalies that satisfy the telFA definition will be found. Thus, no telFA patterns will be missed in the third phase.

### 3.2 Theoretical Analysis

In this section, we present the theoretical analysis of the RAD-fly and RAD-index methods and prove that: (1) both are correct, i.e., each pattern found is teleconnected and satisfies the telFA definition, and (2) both are complete, i.e., all patterns satisfying the telFA definition are found.

**Theorem 1.** *The design decisions “On the Fly” and DN-based are correct, i.e. each pattern  $\langle p, q \rangle$  found by RAD satisfies the telFA definition.*

*Proof.* The pair  $p$  and  $q$  is a teleconnected flow anomaly if both satisfy the following conditions: each satisfies the dpFA definition (Definition 4) and the relationship between  $p$  and  $q$  satisfies the telFA definition (Definition 5). The dpFA patterns  $p$  and  $q$  are found in the first phase using SWEET, a method previously proved correct in [27]. The pattern is then identified as either emerging or vanishing (Phase II). Both  $p$  and  $q$  are neighbors if every time instant in  $p$  is a neighbor of at least one time instant in  $q$ . For each pair of time instants, the actual travel time is found either “On the Fly” (by traversing the path between  $p$  and  $q$ ) or by using the spatio-temporal DN model to identify all the travel times for all paths in the network. A teleconnected flow anomaly is identified in the final phase when each time instant in  $p$  is found to be a neighbor of  $q$ .

**Theorem 2.** *The design decisions “On the Fly” and DN-based are complete, i.e. all teleconnected FA patterns are found by RAD.*

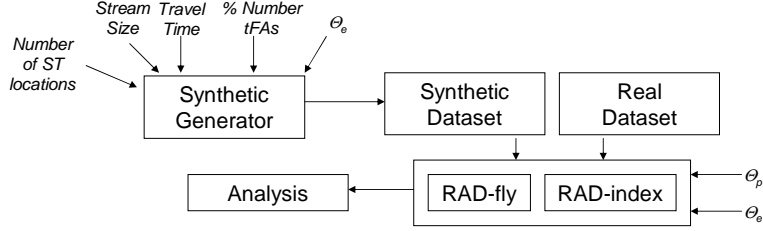


Fig. 5. Experimental Setup

*Proof.* In the first phase of both methods, all dpFAs are found using the SWEET approach [27]. In the second phase, all *emerging* and *vanishing* flow anomalies are found. In the third phase for both methods, only the pair of emerging and vanishing flow anomalies that satisfy the telFA definition will be found. Thus, no telFA patterns will be missed in the second phase for both approaches.

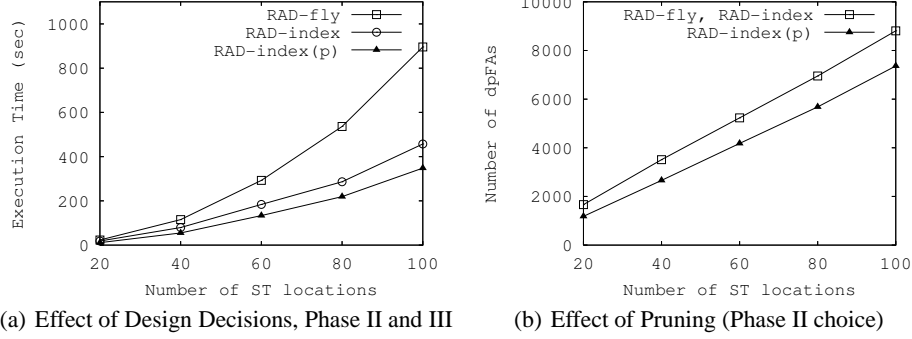
## 4 Experimental Evaluation

In this section, we present our experimental evaluations of our proposed approaches and the workload parameters for our proposed design decisions. We performed our experiments based on the number of nodes in the network and time instants in the series.

**Experimental Setup:** We evaluated the RAD approach using the “On-the-Fly” design decision with no pruning (RAD-fly), the DN-based design decision with no pruning (RAD-index), and the DN-based design decision with pruning (RAD-index(p)). Figure 5 shows the experimental setup. The synthetic generator takes five inputs: (1) the number of ST locations, (2) the length of the time series, (3) the travel time at each node, (4) the percent of tFAs in each time series, and (5) the error threshold to create the synthetic datasets (see Section 4.1). RAD-fly, RAD-index and RAD-index with pruning were analyzed using a generated dataset and a real dataset (measurement of Turbidity). All approaches were compared in terms of execution time and the number of dpFAs found. Execution time was measured based on the system time call in Java before the first phase was executed till the after the third phase was completed. Number of dpFAs was based on the number of flow anomaly patterns found after the second phase of the RAD method. All experiments were performed on an Intel P4 2 GHz 1.2 GB RAM.

### 4.1 Experiments Using Synthetic Data

The synthetic dataset was generated based on the following parameters: (1) the number of ST locations in the network, (2) the size of the time series for each ST location, (3) the percent number of transient flow anomalies across the entire network, (4) the travel time for each ST location, and (5) the error threshold,  $\theta_e$ . Based on these parameters, the generator created a single time series of equal length that was randomly generated and used for each station. The observations in a downstream ST location location was



**Fig. 6.** Phase II and III design decisions over the number of ST locations using synthetic data.

shifted by its specified travel time based on their upstream neighbor. The location of each tFA was chosen randomly and ensured that there will be exactly the percent number of anomalies specified in the input. For experiments to measure the effect on the number of ST locations, the parameters were set as follows: (1) the size of ST locations from 20 to 100, (2) a length of 1000 time instants, (3)  $TT=10$ , (4) 30% tFAs, and  $\Theta_e = 10$ . For the experiments to measure the effect on the size of the time series, the parameters were set as follows: (1) 5 ST locations, (2) a length of 6000 to 30000 time instants, (3)  $TT=10$ , (4) 10% tFAs, and  $\Theta_e = 10$ . The parameters used in this experiment were intended to overlap with those of the real dataset experiments.

**Comparison of Phase II and III design decisions over the ST locations:** Figure 6 gives the results for all three methods; RAD-fly, RAD-index, and RAD-index(p) in terms of the execution time and the number of dpFAs generated after Phase I as the number of ST locations increases. Figure 6a gives the execution time of all three methods. RAD-fly performs more poorly than RAD-index due to the need to compute the travel time between each time instant in the eFAs and vFAs. By contrast, RAD-index uses the spatio-temporal Dynamic Neighborhood (stDN) model to identify the neighborhoods efficiently. The RAD-index(p) method results in further reduction in execution time by removing the *vanishing* flow anomalies that do not have an *emerging* pattern, resulting in fewer dpFAs to analyze in the second phase.

Figure 6b gives the number of dpFAs found after the second phase of each method. RAD-fly and RAD-index give the highest number of dpFAs because there are no filters in the first phase, causing an increase in the number of dpFAs as the number of ST locations increase. RAD-index(p) show a significant reduction in the number of dpFAs after the first phase. This is due to the removal of invalid *vanishing* flow anomalies whose time instants are not neighbors of the first time instant of *emerging* flow anomalies found previously.

**Comparison of Phase II and III design decisions over the time instants:** Figure 7 gives the results for RAD-fly, RAD-index, and RAD-index(p) in terms of the execution time and the number of dpFAs generated after Phase II as the number of time instants increases for each ST location. Figure 7a shows that RAD-fly and RAD-index perform very similarly because there are fewer ST locations in the dataset. However, RAD-

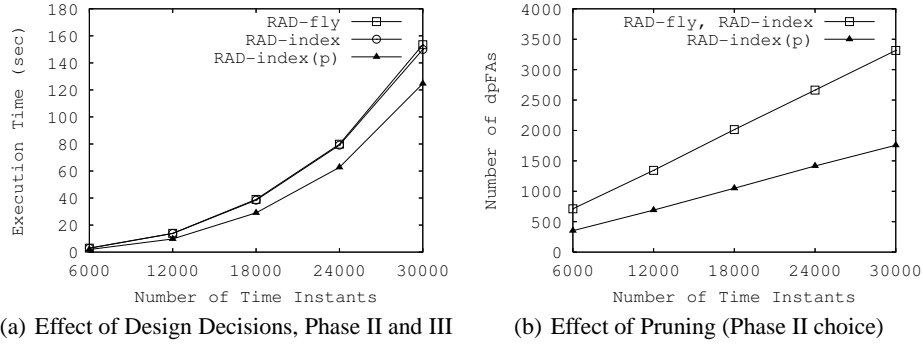


Fig. 7. Phase II and III design decisions over the number of time instants using synthetic data.

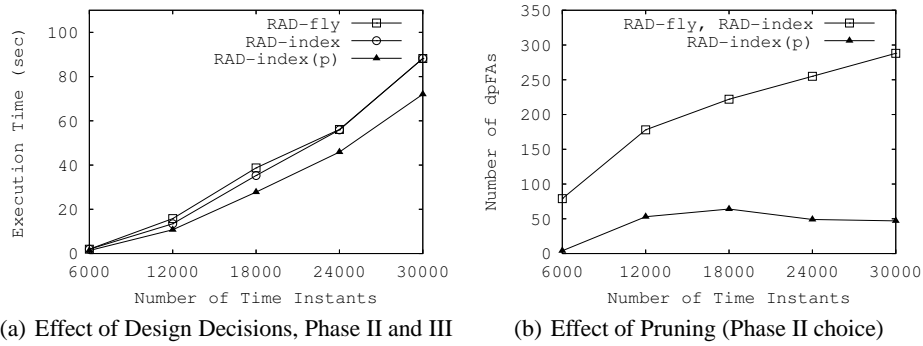


Fig. 8. Phase II and III design decisions over the number of time instants using real data

index(p) outperforms both methods because the pruned *vanishing* flow anomalies result in fewer combinations to compare against in the second phase. Figure 7b gives the number of dpFAs found after Phase I as the number of time instants increases. RAD-index(p) shows fewer dpFAs than the approaches without pruning.

## 4.2 Experiments Using Real Data

The real datasets were obtained from a study site in Shingle Creek, MN where three sensors were placed along a river. The measurement used was turbidity (approximately 30,000 time instants at each sensor). All errors due to sensing problems were removed from the data and the travel time was given. Since the real dataset has only 3 sensors, experiments were performed based on the number of time instants.

**Comparison of Phase II and III design decisions the time instants:** Figure 8 gives the execution time and the number of dpFAs for the real dataset using turbidity for RAD-fly, RAD-index, and RAD-index(p) as the number of time instants increase. Figure 8a gives the execution time for all three proposed methods. As shown in Figure 7a, RAD-index with pruning again performs the fastest. RAD-fly and RAD-index exhibit little difference in execution time, presumably because there were only three ST

locations. Figure 7b shows that RAD-index with pruning produces far fewer dpFAs than the methods without pruning. For both sets of results, RAD-index(p) is more efficient because of the pruning property to eliminate the *vanishing* flow anomalies that do not have an *emerging* neighbor based on the spatio-temporal dynamic neighborhood. It is important to note that the up and down pattern exhibited by RAD-index(p) in Figure 8b is the result of the small flow anomalies collapsing into larger flow anomalies as the time series increase, resulting in various numbers of flow anomalies being pruned.

## 5 Conclusion and Future Work

**Conclusion:** We introduced a novel problem of discovering teleconnected flow anomalies. This problem has a number of important applications for environmental monitoring, video surveillance, and transportation systems. Several new concepts and interest measures were introduced. A RAD approach was proposed that uses novel design decisions of “On the Fly”, spatio-temporal Dynamic neighborhoods based, and a pruning strategy. The proof of correctness and completeness for each proposed method was shown. Experimental evaluation was performed on both synthetic and real datasets.

**Future Work:** The teleconnected flow anomaly problem faces further challenges when the network allows for multiple islands. For example, simply adding one island to a tree network may create two additional paths, two islands may create different paths between nodes, and so, leading to a possible exponential number of time paths between nodes. Thus, future work will investigate the discovery of teleconnected flow anomalies within more complex networks. Further investigation will also be needed to explore alternatives for managing the storage of the spatio-temporal dynamic neighborhoods. Finally, a generalized model will be explored to handle relationships between arbitrary events.

**Acknowledgments:** This work supported by NSF IGERT, USDOD (Public Release Number: 09-341), NSF (EAR 0607138) and USGS/ National Institutes for Water Resources. We would like to thank Kim Koffolt for her comments.

## References

1. Pastor, R.: El niño climate pattern forms in pacific ocean, 2006, [http://www.usatoday.com/weather/climate/2006-09-13-el-nino\\_x.htm](http://www.usatoday.com/weather/climate/2006-09-13-el-nino_x.htm)
2. WFUNA: Millenium project: Global challenges facing humanity (2007)
3. Mason, M.: World’s highest drug levels entering india stream, usa today, 2009, [http://www.usatoday.com/tech/science/environment/2009-01-26-drug-india-stream\\_n.htm](http://www.usatoday.com/tech/science/environment/2009-01-26-drug-india-stream_n.htm)
4. Saulny, S.: Fish-killing virus spreading in the great lakes, new york times, 2007
5. Bruckner, M.: The gulf of mexico dead zone, montana state university, 2008, <http://serc.carleton.edu/microbelife/topics/deadzone/>
6. Matthews, D.A., Effler, S.W., Driscoll, C.T., O’Donnell, S.M., Matthews, C.M.: Electron budgets for the hypolimnion of a recovering urban lake, 1989-2004. *Limnology and Oceanography*, American Society of Limnology and Oceanography **53**(2) (2008) 743–759
7. Hyer, K.E., Hornberger, G.M., Herman, J.S.: Processes controlling the episodic streamwater transport of atrazine and other agrichemicals in the agricultural watershed. *Journal of Hydrology*, Elsevier Science **254** (2001) 47–66

8. Kang, J.M., S.Shekhar, Wennen, C., Novak, P.: Discovering Flow Anomalies: A SWEET Approach. In: IEEE International Conference on Data Mining. (2008) 851–856
9. Amir, A., Apostolico, A., Lewenstein, M.: Inverse pattern matching. *Journal of Algorithms*, Academic Press **24**(2) (1997) 325–339
10. Lee, H., Ng, R.T., Shim, K.: Estimating Rarity and Similarity over Data Stream Windows. In: VLDB. (2007) 195–206
11. Berndt, D.J., Clifford, J.: Using Dynamic Time Warping to Find Patterns in Time Series. In: KDD-94: AAAI Workshop on Knowledge Discovery in Databases. (1994) 359–370
12. Chen, A., Tang, C., an Yuan, C., Peng, J., Hu, J.: Mining Correlations Between Multi-streams Based on Haar Wavelet. In: *Adv. in Computer Science*, Springer LNCS. (2005) 270–271
13. Sayal, M.: Detecting time correlations in time-series data streams. Technical Report HPL-2004-103, Hewlett-Packard Company (2004)
14. Bulut, A., Singh, A.K.: A unified framework for monitoring data streams in real time. In: IEEE ICDE. (2005) 44–75
15. Datar, M., Muthukrishnan, S.: Estimating Rarity and Similarity over Data Stream Windows. In: Annual European symposium on algorithms, Springer LNCS 2461. (2002) 323–334
16. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. In: Proc. of the 9th Intl. Symp. on Spatial and Temporal Databases. (2005) 364–381
17. Spiliopoulou, M., Ntoutsis, I., Theodoridis, Y., Schult, R.: MONIC - Modeling and Monitoring Cluster Transitions. In: ACM SIGKDD. (2006)
18. DeGroot, M., Scheverish, M.J.: Probability and Statistics, 3rd. Ed. Addison Wesley (2002)
19. Knorr, E., Ng, R.: A Unified Notion of Outliers. In: ACM KDD. (1997)
20. Shekhar, S., Lu, C.T., Zhang, P.: A unified approach to spatial outliers detection. *GeoInformatica*, Springer-Verlag **7**(2) (2003) 139–166
21. Li, X., Hodgson, M.E.: Vector field data model and operations. *GIScience and Remote Sensing*, V.H. Winston & Sons **41**(1) (2004) 1–24
22. Zhang, P., Huang, Y., Shekhar, S., Kumar, V.: Exploiting Spatial Autocorrelation to Efficiently Process Correlation-Based Similarity Queries. In: Proceedings of the 8th Intl. Symp. on Spatial and Temporal Databases. (2003) 25–27
23. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective Outbreak Detection in Networks. In: ACM SIGKDD. (2007)
24. Shekhar, S., S.Chawla: *Spatial Databases: A Tour*. Prentice Hall (2002)
25. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* **41** (1959) [www2.informatik.hu-berlin.de/alkox/lehre/lvws0809/verkehr/dijkstra.pdf](http://www2.informatik.hu-berlin.de/alkox/lehre/lvws0809/verkehr/dijkstra.pdf)
26. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* **4**(2) (1968)
27. Kang, J.M., S.Shekhar, Wennen, C., Novak, P.: Discovering Flow Anomalies: A SWEET Approach. In: University of Minnesota, MN, Technical Report, 09-006. (2009)