Prediction-Based Geometric Monitoring Over Distributed Data Streams [*]

Nikos Giatrakos[†]
Dept. of Informatics
University of Piraeus
ngiatrak@unipi.gr

Antonios Deligiannakis
Dept. of Electronic and
Computer Engineering
Technical University of Crete
adeli@softnet.tuc.gr

Minos Garofalakis
Dept. of Electronic and
Computer Engineering
Technical University of Crete
minos@softnet.tuc.gr

Izchak Sharfman
Faculty of Computer Science
Technion
tsachis@cs.technion.ac.il

Assaf Schuster
Faculty of Computer Science
Technion
assaf@cs.technion.ac.il

ABSTRACT

Many modern streaming applications, such as online analysis of financial, network, sensor and other forms of data are inherently distributed in nature. An important query type that is the focal point in such application scenarios regards actuation queries, where proper action is dictated based on a trigger condition placed upon the current value that a monitored function receives. Recent work [18, 20, 21] studies the problem of (non-linear) sophisticated function tracking in a distributive manner. The main concept behind the geometric monitoring approach proposed there, is for each distributed site to perform the function monitoring over an appropriate subset of the input domain. In the current work, we examine whether the distributed monitoring mechanism can become more efficient, in terms of the number of communicated messages, by extending the geometric monitoring framework to utilize prediction models. We initially describe a number of local estimators (predictors) that are useful for the applications that we consider and which have already been shown particularly useful in past work. We then demonstrate the feasibility of incorporating predictors in the geometric monitoring framework and show that prediction-based geometric monitoring in fact generalizes the original geometric monitoring framework. We propose a large variety of different prediction-based monitoring models for the distributed threshold monitoring of complex functions. Our extensive experimentation with a variety of real data sets, functions and parameter settings indicates that our approaches can provide significant communication savings ranging between two times and up to three orders of magnitude, compared to the transmission cost of the original monitoring framework.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous;
H.2.4 [**Database Management**]: Systems - Query processing

Keywords

Continuous distributed monitoring, data streams, prediction models

1. INTRODUCTION

A wide variety of modern applications relies on the *continuous* processing of vast amounts of arriving data in order to support decision making procedures in real time. Examples include network administration, stock market analysis, environmental, surveillance and other application scenarios. These settings are, more often than not, inherently *distributed* in nature. For instance, consider the case of a network operation center where data is produced by hundreds or thousands of routers [2, 5, 4] or the case of environmental as well as control applications where wireless sensor network adoption has become of great importance [15].

Due to the distributed nature of data production in the aforementioned scenarios, the major challenge confronted by algorithms dealing with their manipulation is to reduce communication [2, 5, 4, 18, 20, 21, 8]. This happens because the central collection of data is not feasible in large scale applications. Furthermore, in the case of sensor network deployments, central data accumulation results in depleting the power supply of individual sensors reducing the network lifetime [15].

An important query type that is of the essence in the aforementioned fields regards the monitoring of a trigger condition defined upon the range of values a function of interest receives [18, 20, 21, 10, 12, 13, 11]. For instance, in order to perform spam detection on a number of dispersed mail servers, algorithms base their decisions on whether the value of the information gain function globally exceeds a given threshold [20]. Moreover, in the example of the network operation center, denial of service attacks are detected by attempting to pinpoint strangely high (based on a given threshold) number of distinct source addresses routing packets across various destinations within the network [8].

Recently, the work in [18, 20] was the first that introduced a generic paradigm for monitoring complex (non-linear) functions defined over the average of local vectors maintained at distributed sites. Their proposed geometric approach essentially monitors the area of the input domain where the average vector may lie, rather

than monitoring the function's value itself. The monitoring is performed in a distributed manner, by assigning each node a monitoring zone, expressed as a hypersphere, which is nothing more than a subset of the input domain where the average vector may lie. Communication is shown to be necessary only if at least one site considers it likely that the condition of the monitored function may have changed since the last communication between the sites.

In this work, we examine the potentials of a simple (yet powerful), easy to locally maintain approach in order to further reduce transmissions towards the central source. In particular, we foster prediction models so as to describe the evolution of local streams. The adoption of prediction models has already been proven beneficial in terms of bandwidth preservation [2, 5, 4] in distributed settings. Initially, we extend the geometric monitoring framework of [18, 20] and illustrate how it can incorporate predictors, in order to forecast the evolution of local data vectors of sites. We exhibit the way the geometric monitoring framework is modified to encompass constructed predictors and identify the peculiarities occurred upon predictors' adoption. In contrast to the ascertainment of prior works [2, 5, 4], we prove that the mere utilization of local predictions is hardly adequate to guarantee communication preservation even when predictors are quite capable of describing local stream distributions. We then proceed by establishing a theoretically solid monitoring framework that incorporates conditions managing to guarantee fewer contacts with the central source. Eventually, we develop a number of mechanisms, along with extensive speculative analysis, that relax the previously introduced framework, base their function on simpler criteria, and in practice yield significant transmission reduction. Our main contributions are:

- We introduce the adoption of prediction models in the setting of tracking complex, non-linear functions utilizing the geometric approach [18, 20]. We exhibit the way prediction models can be locally adopted by sites and we show the characteristics they attribute to the geometric approach. We then illustrate that the initial geometric monitoring framework of [18, 20] is a special case of our, more general, prediction-based geometric monitoring framework.

- We point out the failure of conventional notions of good predictors to be applied in this setting and manage to establish a solid theoretic framework consisting of sufficient conditions that do render prediction models capable of guaranteeing reduced bandwidth consumption.

- We expose a number of novel tracking mechanisms relaxing the previously (hard to verify in a distributed manner) identified sufficient conditions. Using the simplest possible primitives regarding prediction models' behavior, we thoroughly study the potentials of our new tracking techniques to achieve communication preservation.

- We present an extensive experimental analysis using a variety of real data sets, parameters and functions of interest. Our evaluation shows that our approaches can provide significant communication load reduction with savings ranging from 2 times and in some cases reaching 3 orders of magnitude compared to the transmission cost of the original bounding algorithm.

2. RELATED WORK

Recently, substantial efforts have been devoted on tracking and querying distributed data streams [3]. The geometric monitoring framework which is leveraged by our approaches was introduced in [18, 20] and was later enhanced in [21]. The optimizations proposed in [21] are orthogonal to our approaches, but note that the techniques of [21] either require data to conform with a multivariate

normal distribution or entail a number of solutions to a series of optimization problems that may increase the computational load. The latter renders their adoption unaffordable in resource constraint environments such as [19]. On the contrary, our approaches are based on simple predictors' adoption that remain adaptable to changing data distributions and are easy to maintain even when resource constraints exist. In other work related to the geometric monitoring approach, [19] discusses an application of the framework of [18, 20] to clustered sensor network settings. The more recent work of [17] adopts the geometric approach and proposes a tentative bound algorithm to monitor threshold queries in distributed databases (rather than distributed data streams) for functions with bounded deviation.

Prediction models in the context of distributed data streams have already been fostered in previous work to monitor one-dimensional quantiles [5] and randomized sketch summaries [4]. Their adoption has been proven beneficial in terms of reducing the communication burden. Contrary to previous approaches our focus is on the benefits they can provide in the context of the geometric monitoring framework for tracking non-linear threshold functions.

In related work regarding distributed trigger monitoring, [13] provides a framework for monitoring thresholded counts over distributed data streams, while [12] designs techniques that decompose the problem of detecting when the sum of a distributed set of variables exceeds a given threshold. Based on [12] anomaly detection techniques are studied in [10] and [11]. The recent work of [6] provides upper and lower communication bounds for approximate monitoring of thresholded F_p moments, with $p = 0, 1, 2$.

Other works focus on tracking specific types of functions over distributed data streams. The work of [16] considers simple aggregation queries over multiple sources, while [1] focuses on monitoring top-k values. Furthermore, [8] distributively monitors set-expression cardinalities using a scheme for charging local changes against single site's error tolerance. [22] considers the problem of tracking heavy hitters and quantiles in a distributed manner establishing optimal algorithms to accomplish the task. Eventually, [7] studies the problem of clustering distributed data streams, while [23] generalizes the previous approach to hierarchical environments.

3. PRELIMINARIES

In this section we first provide helpful background work related to function monitoring using the geometric approach. We then describe local stream predictors, which have been utilized in past work. The notation used in this paper appears in Table 1.

3.1 The Geometric Monitoring Framework

As in previous works [5, 20, 2, 4, 21], we assume a distributed, two-tiered setting, where data arrives continuously at n geographically dispersed sites. At the top tier, a central coordinator exists that is capable of communicating with every site, while pairwise site communication is only allowed via the coordinating source.

Each site S_i, $i \in [1..n]$ participating at the bottom tier receives updates on its local stream and maintains a d-dimensional local measurements vector $v_i(t)$. The *global measurements vector* $v(t)$ at any given timestamp t, is calculated as the weighted average of $v_i(t)$ vectors, $v(t) = \frac{\sum_{i=1}^{n} w_i v_i(t)}{\sum_{i=1}^{n} w_i}$, where $w_i \geq 0$ refers to the weight associated with a site. Usually, w_i corresponds to the number of data points received by S_i [18]. Our aim is to continuously monitor whether the value of a function $f(v(t))$, defined upon $v(t)$, lies above/below a given threshold T. We use the term *threshold surface* to denote the area of the input domain where $f(v(t)) = T$.

During the monitoring task using the geometric approach [18,

Table 1: Notation used

Symbol	Description
n	The number of sites
S_i	The i-th site
t_s	Timestamp of the last synchronization
$v(t)$	Global measurements vector at time t ($\sum_{i=1}^{n} w_i v_i(t) / \sum_{i=1}^{n} w_i$)
$e(t)$	Estimate vector at time t (equal to $v(t_s)$)
$e^p(t)$	The predicted estimate vector ($\sum_{i=1}^{n} w_i v_i^p(t) / \sum_{i=1}^{n} w_i$)
$v_i(t)$	Local measurements vector at S_i at time t
w_i	Number of data points at S_i
$u_i(t)$	Drift vector (equals to $e(t) + v_i(t) - v_i(t_s)$)
$v_i^p(t)$	Local predictor of S_i at time t
$u_i^p(t)$	Prediction deviation vector ($e^p(t) + v_i(t) - v_i^p(t)$)
$B_c^{\|r\|}$	Local constraint (ball) centered at c with radius $\|r\|$

20], the coordinator may request that all sites transmit their local measurements vectors and subsequently calculates $v(t)$, performs the required check on $f(v(t))$, and transmits the $v(t)$ vector to all sites. The previous process is referred to as a *synchronization* step. Let $v_i(t_s)$ denote the local measurements vector that S_i communicated during the last synchronization process at time t_s. The global measurements vector computed during a synchronization step is denoted as the *estimate vector e*, where $e = \sum_{i=1}^{n} w_i v_i(t_s) / \sum_{i=1}^{n} w_i$.

After a synchronization, sites keep up receiving updates of their local streams and accordingly maintain their $v_i(t)$ vectors. At any given timestamp, each site S_i individually computes $v_i(t) - v_i(t_s)$ and the local *drift vector* $u_i(t) = e + (v_i(t) - v_i(t_s))$. Since

$$v(t) = \frac{\sum_{i=1}^{n} w_i v_i(t)}{\sum_{i=1}^{n} w_i} = e + \frac{\sum_{i=1}^{n} w_i (v_i(t) - v_i(t_s))}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} w_i u_i(t)}{\sum_{i=1}^{n} w_i}$$

$v(t)$ constitutes a convex combination of the drift vectors. Consequently, $v(t)$ will always lie in the convex hull formed by the $u_i(t)$ vectors: $v(t) \in Conv(u_1(t), \ldots, u_n(t))$, as depicted in Figure 1.

Please note that each site can compute the last known value of the monitored function as $f(e)$ and can, thus, determine whether this value lies above/below the threshold T. Since $v(t) \in Conv(u_1(t), \ldots, u_n(t))$, if the value of the monitored function in the entire convex hull lies in the same direction (above/below the threshold T) as $f(e)$, then it is guaranteed that $f(v(t))$ will lie in that side. In this case, the function will certainly not have crossed the threshold surface. The key question is: "how can the sites check the value of the monitored function in the entire convex hull, since each site is unaware of the current drift vectors of the other sites"? This test can be distributively performed as described in Theorem 1, while an example (in 2-dimensions) is included in Figure 1.

THEOREM 1. *[18, 20] Let $x, y_1, \ldots, y_n \in R^d$ be a set of d dimensional vectors. Let $Conv(x, y_1, \ldots, y_n)$ be the convex hull of $x, y_1, \ldots, y_n$. Let $B_{\frac{x+y_i}{2}}^{\|\frac{x-y_i}{2}\|}$ be a ball centered at $\frac{x+y_i}{2}$ with a radius of $\|\frac{x-y_i}{2}\|$ that is, $B_{\frac{x+y_i}{2}}^{\|\frac{x-y_i}{2}\|} = \{z \in R^d : \|z - \frac{x+y_i}{2}\| \leq \|\frac{x-y_i}{2}\|\}$. Then, $Conv(x, y_1, \ldots, y_n) \subset \bigcup_{i=1}^{n} B_{\frac{x+y_i}{2}}^{\|\frac{x-y_i}{2}\|}$.*

With respect to our previous discussion x corresponds to e while y_i vectors refer to the drift vectors $u_i(t)$. Hence, sites need to compute their local constraints in the form of $B_{\frac{e+u_i(t)}{2}}^{\|\frac{e-u_i(t)}{2}\|}$ and independ-
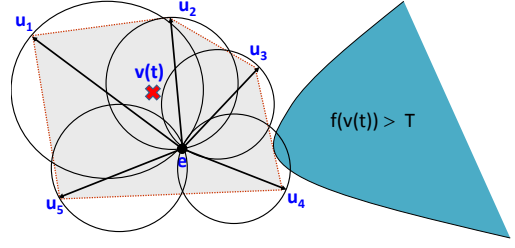

Figure 1: Demonstration of the geometric framework rationale. *Conv* $(u_1, \ldots, u_n)$ **is depicted in gray, while the actual position of e and the current $v(t)$ are shown as well. Black spheres refer to the local constraints constructed by sites to assess possible threshold crossing. $v(t)$ is guaranteed to lie within the union of these locally constructed spheres. Since one of the spheres crosses the threshold surface, $f(v(t))$ and $f(e)$ may not lie at the same side relative to the threshold T. Hence, a synchronization needs to be performed.**

dently check whether a point within these balls may cause a threshold crossing. If this indeed is the case, a synchronization step takes place. Note that since *Conv* $(e, u_1, \ldots, u_n)$ is a subset of the union of local ball constraints, the framework may produce a synchronization in cases where the convex hull has not actually crossed the threshold surface (false positives).

In summary, each site in the geometric monitoring framework manages to track a subset of the input domain. The overall approach achieves communication savings since the coordinator needs to collect the local measurement vectors of the sites only when a site locally detects (in its monitored area of the input domain) that a threshold crossing may have occurred.

3.2 Local Stream Predictors

We now outline the properties of some prediction model options that have already been proven useful in the context of distributed data streams [4, 5]. Please note, beforehand, that the concept of their adoption is to keep such models as simple as possible, and yet powerful enough to describe local stream distributions. It can easily be conceived that more complex model descriptors can be utilized, which however incur extra communication burden when sites need to contact the coordinating source [4, 5]. In our setting, this translates to an increased data transmission overhead during each synchronization step. In our discussion, hereafter, we utilize the term *predictor* to denote a prediction estimator for future values of a local measurements vector. Using a similar notation to the one of Section 3.1, we employ $v_i^p(t)$ to denote the prediction for the local measurements vector of site S_i at timestamp t.

The Static Predictor. The simplest guess a site may take regarding the evolution of its local measurements vector is that its coordinates will remain unchanged with respect to the values they possessed in the last synchronization: $v_i^p(t) = v_i(t_s)$. It is also evident that this predictor is trivial to maintain in both the sites and the coordinator. Moreover, it requires no additional information to be communicated towards the coordinator upon a synchronization step. Using

Table 2: Local Stream Predictors' Summary

Predictor	Info.	Pred. Local Vector (v_i^p)
Static	$\emptyset$	$v_i(t_s)$
Linear Growth	$\emptyset$	$\frac{t}{t_s} v_i(t_s)$
Velocity/Acceleration	vel_i	$v_i(t_s) + (t - t_s)vel_i + (t - t_s)^2 accel_i$

the static predictor, in the absence of a synchronization step, the coordinator estimates that $v(t) = e$.

The static predictor may be a good choice only in settings where the evolution of the values in each local measurements vector is unpredictable, or local measurements vectors change rarely.

The Linear Growth Predictor. The next simple, but less restrictive, assumption that can be made is that local vectors will scale proportionally with time. In particular, $v_i^p(t) = \frac{t}{t_s} v_i(t_s)$ which is the only calculation individual sites and the coordinating source need to perform in order to derive an estimation of $v_i(t)$ at any given time. Please note that, using this predictor, the best guess that a coordinator can make for the value of $v(t)$ is equal to $\frac{t}{t_s} v(t_s) = \frac{t}{t_s} e$. As with the Static Predictor, the Linear Growth Predictor requires no additional information to be transmitted upon a synchronization.

We can deduce that the Linear Growth Predictor is built on the assumption that $v_i(t)$ vectors evolve, but that their evolution involves no direction alterations. Consequently, it can be adopted so as to approximate local streams in which $v_i(t)$ vectors' coordinates are expected to uniformly increase by a time dependent factor.

The Velocity/Acceleration Predictor. The Velocity/Acceleration (*VA*) Predictor is a much more expressive predictor. VA employs additional vectors that attempt to capture both the scaling and directional change that $v_i(t)$ may undertake. More precisely, in the *VA* predictor the future value of the local measurements vector is estimated as $v_i^p(t) = v_i(t_s) + (t - t_s)vel_i + (t - t_s)^2 accel_i$. Since the velocity vel_i and the acceleration $accel_i$ of the local stream are capable of expressing both possible types of $v_i(t)$ alterations, it provides an enriched way to approximate its behavioral pattern.

In a way similar to [4], when a synchronization is about to take place, S_i is required to compute the velocity vector vel_i utilizing a window of the W most recent updates it received. Given that window, the velocity vector can be calculated by computing the overall disposition as the difference between $v_i(t)$ and the local vector instance corresponding to the first position of the window.[1] Scaling this outcome by the time difference between the window extremes provides vel_i. In addition, the $accel_i$ value can be computed as the difference between the current velocity and corresponding velocity calculated in the previous synchronization. Scaling the previous result by $1/(t - t_s)$ computes a proper $accel_i$ vector. Additional approaches based on use of vel_i and $accel_i$ values can be found in [4].

It is easy to see that the flexibility provided by the *VA* Predictor comes at the cost of the transmission of vel_i (along with $v_i(t)$) during each synchronization. We note that $accel_i$ does not need to be communicated to the coordinator, since the coordinator is already aware of the previously computed velocity vectors of each site.

Table 2 summarizes the described predictor characteristics. It is important to emphasize that the prediction-based monitoring framework described in the next sections can utilize any predictor and is, thus, not restricted to the predictors presented in this section.

4. PREDICTION-BASED MONITORING

In this section we first motivate the need to incorporate predictors in the geometric monitoring framework and then demonstrate how this can be achieved. We then illustrate that the initial geometric monitoring framework of [18, 20] is a special case of our, more general, prediction-based geometric monitoring framework. Subsequently, we define the notion of a good predictor and demonstrate that good predictors lead to monitoring a smaller subset of

---

[1]Please note that each update may not arrive at each timestamp. Thus, the timestamp of the first update in the window may in general be different than $t - W + 1$.

the domain space, thus potentially leading to fewer synchronizations and, hence, fewer transmitted messages.

Motivation for Predictors. Figure 1 demonstrates a motivating example of why it may be beneficial to incorporate predictors in the geometric monitoring framework. In the illustrated example, the sphere of u_4 has crossed the threshold surface. Based on their definition, the direction of each drift vector essentially depicts how the values of the corresponding local measurements vector have changed since the last synchronization. Using the geometric monitoring approach, a synchronization will take place because S_4 will detect a threshold crossing. A plausible question is: "Could we avoid such a synchronization step, if the changes in the values of the five local measurements vectors could have been predicted fairly accurate"? For example, if we could have predicted the change (drift) in the local measurements vectors of each site fairly accurately, then we would have determined that $v(t)$ has probably not moved closer to the threshold surface and, thus, avoid the synchronization step. The above example motivates the need for prediction-based geometric monitoring.

How to Incorporate Predictors. As explained in Section 3.2, the coordinator can receive, during a synchronization step, information regarding the predicted local measurements vector $v_i^p(t)$ of each site. Thus, the coordinator will be able to compute an estimation of $v(t)$ provided by the local predictors as: $e^p(t) = \frac{\sum_{i=1}^{n} w_i v_i^p(t)}{\sum_{i=1}^{n} w_i}$, which we will term as the *predicted estimate vector*. Based on $e^p(t)$, we now show that the coordinator can continuously check the potential threshold crossings. However, in this case a synchronization is required only when $e^p(t)$ and $v(t)$ are likely to be placed in different sides of the threshold surface.

In the context of the geometric monitoring framework, we first observe that:

$$v(t) = \frac{\sum_{i=1}^{n} w_i v_i(t)}{\sum_{i=1}^{n} w_i} = e^p(t) + \frac{\sum_{i=1}^{n} w_i(v_i(t) - v_i^p(t))}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} w_i u_i^p(t)}{\sum_{i=1}^{n} w_i}$$

where $u_i^p(t) = e^p(t) + (v_i(t) - v_i^p(t))$ denotes the vector expressing the *prediction deviation*. Thus, similar to our analysis in Section 3.1, $v(t) \in Conv\ (u_1^p(t), \ldots, u_n^p(t)) \subset \bigcup_{i=1}^{n} B \left\| \frac{e^p(t) - u_i^p(t)}{2} \right\|_{\frac{e^p(t) + u_i^p(t)}{2}}$. Since $v(t)$ lies in the convex hull $Conv(u_1^p(t), \ldots, u_n^p(t))$, each site S_i can monitor the ball that has as endpoints of its diameter the estimated predicted vector $e^p(t)$ and its prediction deviation $u_i^p(t)$.

Please note that the geometric monitoring approach of [18, 20] corresponds to utilizing a static predictor (this leads to $v_i^p(t) = v(t_s)$, $e^p(t) = e$ and $u_i^p(t) = u_i(t)$) and is, thus, a special case of our, more general, prediction-based monitoring framework.

Defining a Good Predictor. Upon utilizing a predictor, as long as local forecasts ($v_i^p(t)$) remain sound, we expect that they will approximate the true local vectors $v_i(t)$ to a satisfactory degree at any given timestamp. This means that each $v_i^p(t)$ will be in constant proximity to the $v_i(t)$ vector, when compared to $v_i(t_s)$. Formally:

PROPERTY 1. *A **Good Predictor** possesses the property:*

$$\|v_i(t) - v_i^p(t)\| \leq \|v_i(t) - v_i(t_s)\|\ \forall t \geq t_s$$

Property 1 lies, implicitly or not, in the core of predictors' adoption in distributed stream settings. It expresses the notion of a useful, in terms of bandwidth consumption reduction, predictor present

in previous works [2, 4, 5] which have managed to exhibit important improvements by exploiting the above fact. Hence, we start by exploring the benefits of the notion of good predictors expressed by Property 1 within the geometric monitoring setting.

Predictors satisfying Property 1 yield stricter local constraints for the bounding algorithm compared to the original monitoring mechanism (Section 3.1). This happens because $\|v_i(t) - v_i^p(t)\| \leq \|v_i(t) - v_i(t_s)\| \Leftrightarrow \|u_i^p(t) - e^p(t)\| \leq \|u_i(t) - e\|$ and the radius of the constructed balls will always be smaller. An example of prediction-based monitoring is depicted in Figure 2.

Consequently, a good predictor results in the sites monitoring a tighter convex hull, namely $Conv\ (u_1^p(t), \ldots, u_n^p(t))$, than the corresponding convex hull of the original geometric monitoring framework. This yields the construction of tighter local constraints and, as already mentioned, a synchronization is required only when $e^p(t)$ is likely to be placed in a different side of the threshold surface to the one of $v(t)$. Hence, a synchronization is again caused when any ball $B^{\left\|\frac{e^p(t)-u_i^p(t)}{2}\right\|}_{\frac{e^p(t)+u_i^p(t)}{2}}$ crosses the threshold surface.

Despite the fact that this mechanism may in practice be useful, it cannot guarantee fewer synchronizations because $Conv\ (u_1^p(t), \ldots, u_n^p(t))$, although tighter, might still be placed closer than $Conv\ (u_1(t), \ldots, u_n(t))$ to the threshold surface. This in turn will cause some $B^{\left\|\frac{e^p(t)-u_i^p(t)}{2}\right\|}_{\frac{e^p(t)+u_i^p(t)}{2}}$ to cross the threshold before any $B^{\left\|\frac{e-u_i(t)}{2}\right\|}_{\frac{e+u_i(t)}{2}}$ does (Figure 2). This observation shows that the conventional concept of good predictors fails to adapt in the current setting since it does not guarantee by itself fewer synchronizations.

5. STRONG MONITORING MODELS

The concluding observations of Section 4 raise a concern regarding the sufficient conditions that should be fulfilled for the predictors to always yield fewer synchronizations than the original framework. Apparently, this happens when the surface of the monitoring framework devised by the predictors is contained inside the monitored surface of the original framework. In other words, we need to define the prerequisites for constructing local constraints that are always included in $\bigcup_{i=1}^{n} B^{\left\|\frac{e-u_i(t)}{2}\right\|}_{\frac{e+u_i(t)}{2}}$ utilized by the original framework. Let $Sur(P)$ be the surface monitored by any alternative mechanism that adopts predictors while operating. A monitoring model is defined as *strong* if the following property holds:

PROPERTY 2. *A* **Strong** *predictor-based* **Monitoring Model** *possesses the property:* $Sur(P) \subseteq \bigcup_{i=1}^{n} B^{\left\|\frac{e-u_i(t)}{2}\right\|}_{\frac{e+u_i(t)}{2}}$

5.1 Containment of Convex Hulls

According to Theorem 1, after computing the local drift vectors and prediction deviations, we are free to choose any common, *reference vector* in order to perform the monitoring task. Thus, it is not mandatory for the sites to use e and $e^p(t)$ as a common reference point in order to construct their monitoring zones. In fact, the sites could use any common point as an endpoint of the diameter of their monitoring zones.

An important observation that we prove in this section is that a predictor-based monitoring model satisfies Property 2 when (1) every prediction deviation vector is contained in the convex hull of the estimate vector and the drift vectors defined by the original bounding algorithm, and (2) an appropriate reference vector is selected.

Before proving our observation, we first show that for any triplet of vectors $z, y, x \in R^d$, the condition $z \in B^{\left\|\frac{y-x}{2}\right\|}_{\frac{y+x}{2}}$ is equivalent to
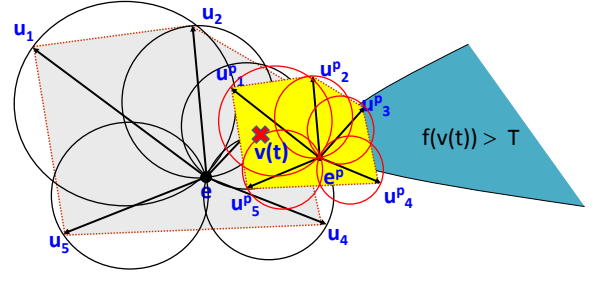

Figure 2: The red balls demonstrate the local constraints of sites when using a sample good predictor. A good predictor results in the tighter convex hull $Conv\ (u_1^p(t), \ldots, u_n^p(t))$ (depicted in yellow). Here, fewer synchronizations are not guaranteed, since $\bigcup_{i=1}^{n} B^{\left\|\frac{e^p(t)-u_i^p(t)}{2}\right\|}_{\frac{e^p(t)+u_i^p(t)}{2}}$ crosses the threshold before $\bigcup_{i=1}^{n} B^{\left\|\frac{e-u_i(t)}{2}\right\|}_{\frac{e+u_i(t)}{2}}$.

$\langle x - z, y - z \rangle \leq 0$ where the notation $\langle ., . \rangle$ refers to the inner product of two vectors. Whenever it is appropriate, we omit the temporal reference symbol (t) in the vectors to simplify the exposition.

LEMMA 1. $z \in B^{\left\|\frac{y-x}{2}\right\|}_{\frac{y+x}{2}}$ *if and only if* $\langle x - z, y - z \rangle \leq 0$.

PROOF. Recall that if $z \in B^{\left\|\frac{y-x}{2}\right\|}_{\frac{y+x}{2}}$, then $\|z - \frac{x+y}{2}\| \leq \|\frac{x-y}{2}\|$. This is equivalent to $\frac{1}{4}\langle 2z - (x+y),\ 2z - (x+y)\rangle - \frac{1}{4}\langle x-y,\ x-y\rangle \leq 0$. Recall that the inner product is distributive, i.e. $\langle a+b,\ c\rangle = \langle a,\ c\rangle + \langle b,\ c\rangle$, and symmetric, i.e. $\langle a,\ b\rangle = \langle b,\ a\rangle$ Therefore: $\frac{1}{4}\langle 2z - (x+y), 2z - (x+y)\rangle - \frac{1}{4}\langle x-y, x-y\rangle = \frac{1}{4}\langle (z-x) + (z-y), (z-x) + (z-y)\rangle - \frac{1}{4}\langle (z-x) - (z-y), (z-x) - (z-y)\rangle = \langle z-x, z-y\rangle = \langle x-z, y-z\rangle$. $\square$

We now proceed to prove in Lemma 2 that a predictor-based monitoring model that maintains each prediction deviation vector contained in the convex hull of the drift vectors defined by the original bounding algorithm is a strong predictor-based monitoring model if it also selects the same reference vector (e.g., e instead of e^p) as the original framework. A direct result is that the area monitored by the sites is a subset of the corresponding area of the original framework. This, in turn, leads to fewer synchronizations, since every time a site detects a potential threshold crossing in the predictor-based monitoring model, at least one site would also have detected the same threshold crossing (for the same vector of the input domain) in the original framework.

LEMMA 2. *Let* $u_i^p \in Conv(u_1, ..., u_n)\ \forall i \in \{1..n\}$. *Then*

$$B^{\left\|\frac{e-u_i^p}{2}\right\|}_{\frac{e+u_i^p}{2}} \subseteq \bigcup_{i=1}^{n} B^{\left\|\frac{e-u_i}{2}\right\|}_{\frac{e+u_i}{2}}.$$

PROOF. For each $u_i^p \in Conv(u_1, ..., u_n)$ there exist $\lambda_1, \lambda_2, \ldots, \lambda_n$ such that $\lambda_i \geqslant 0\ (i \in \{1..n\})$, $\sum_{i=1}^{n} \lambda_i = 1$ and $u_i^p = \sum_{i=1}^{n} \lambda_i u_i$. Let $h \in B^{\left\|\frac{e-u_i^p}{2}\right\|}_{\frac{e+u_i^p}{2}}$. We will show that for at least one of the u_i vectors, $h \in B^{\left\|\frac{e-u_i}{2}\right\|}_{\frac{e+u_i}{2}}$. According to Lemma 1: $\langle h-e, h-u_i^p\rangle \leq 0$. Therefore:

$$\langle h-e, h-u_i^p\rangle = \langle h-e, \sum \lambda_i h - \sum \lambda_i u_i\rangle = $$
$$\langle h-e, \sum \lambda_i (h-u_i)\rangle = \sum \lambda_i \langle h-e, h-u_i\rangle \leq 0$$

Since $\lambda_i \geqslant 0$, it follows that for at least one u_i with $\lambda_i > 0$, $\langle h - e, h - u_i \rangle \leq 0$, which implies (Lemma 1) that $h \in B_{\frac{e+u_i}{2}}^{\left\| \frac{e-u_i}{2} \right\|}$. $\quad\square$

A trivial example of a strong predictor-based monitoring model is the static predictor which, as mentioned in Section 4, is equivalent to the original framework of [18, 20].

Unfortunately, the containment constraints are not easily abided by any other chosen predictor and, even if they are, it appears hard to dictate a way that allows sites to distributively identify that fact. We will revisit the convex hull containment issues in Section 6.1.

5.2 Convex Hull Intersection Monitoring

An important observation that we make is that, as $v(t) \in Conv(u_1, \dots, u_n)$ and $v(t) \in Conv(u_1^p, \dots, u_n^p)$, these two convex hulls cannot be disjoint (Fig. 2). One could, thus, seek ways to exploit this fact, which limits the possible locations of $v(t)$, in order to reduce the size of the monitoring zones of each site which, in turn, will potentially lead to fewer detected threshold crossings. We thus seek to come up with new local constraints in the context of predictor-based monitoring models that cover the intersection of the two convex hulls and which also fulfill Property 2. To proceed towards that goal we first formally formulate an enhanced version of Property 1.

PROPERTY 3. *A **Universally Good Predictor** possesses the property:* $\|v_i(t) - v_i^p(t)\| \leq \|v_j(t) - v_j(t_s)\|$ *for any pair of sites* S_i, S_j

In other words for universally good predictors:

$$\min_{k=1..n} \|e - u_k\| \geq \max_{k=1..n} \|e^p - u_k^p\| \qquad (1)$$

Property 3 yields $\|u_i^p - e^p\| \leq \|u_j - e\|$ for any pair of sites S_i, S_j. The latter result is produced by simply adding as well as subtracting e^p, e to the left and right side of its inequality, respectively. The following lemma utilizes this fact to devise appropriate local constraints to be fostered at each site S_i.

LEMMA 3. *If Property 3 holds, each site S_i needs to examine whether* $B_{\frac{e+u_i}{2}}^{\left\| \frac{e-u_i}{2} \right\|} \cap B_{e^p}^{\|e-u_i\|}$ *crosses the threshold, since:*

$$Conv(u_1, \dots, u_n) \cap Conv(u_1^p, \dots, u_n^p) \subset \bigcup_{k=1}^{n} B_{\frac{e+u_k}{2}}^{\left\| \frac{e-u_k}{2} \right\|} \cap B_{e^p}^{\|e-u_k\|}$$

PROOF. We will demonstrate that any vector $h \in R^d$ which lies in $Conv(u_1, \dots, u_n) \cap Conv(u_1^p, \dots, u_n^p)$ is also included in at least one intersection $B_{\frac{e+u_k}{2}}^{\left\| \frac{e-u_k}{2} \right\|} \cap B_{e^p}^{\|e-u_k\|}$ of a site S_k ($k \in \{1..n\}$). What is certain is that, due to Property 3, $h \in Conv(u_1^p, \dots, u_n^p) \Rightarrow h \in \bigcup_{k=1}^{n} B_{\frac{e^p+u_k^p}{2}}^{\left\| \frac{e^p-u_k^p}{2} \right\|} \Rightarrow h \in \bigcup_{k=1}^{n} B_{e^p}^{\|e^p-u_k^p\|} \Rightarrow h \in B_{e^p}^{\|e-u_k\|}$. Since h is definitely contained as well in at least one of the balls $B_{\frac{e+u_k}{2}}^{\left\| \frac{e-u_k}{2} \right\|}$ (Theorem 1) constructed by the sites, which implies that h will be examined by at least one site. The proof follows immediately. $\quad\square$

According to Lemma 3, universally good predictors guarantee Property 2, thus leading to a decreased synchronization frequency. Nonetheless, in practice we cannot safely assume that predictors are always universally good. Hence, sites need to constantly check Inequality 1. This check can only be done by gathering all $\|e - u_i\|$, $\|e^p - u_i^p\|$ values to the coordinator so as to compute the respective minimum and maximum. But if we allow that, the number of messages will be equivalent to that of continuous, central data collection. Thus, in Section 6.2, we seek to devise alternative implementations for intersection monitoring that employ more relaxed conditions and call for a synchronization only when a violation of newly devised local constraints occurs.

6. SIMPLIFIED ALTERNATIVES

6.1 Relaxing the Containment Condition

The containment of convex hulls (Section 5.1), as a sufficient prerequisite to achieve accordance with Property 2 is seemingly hard to achieve, let alone come up with ways to continuously check it in a distributive manner. To confront the above drawbacks we investigate an alternative approach which relaxes that condition. Instead of distributively checking the containment condition, we direct our interest to the more practical alternative of making it likely. Intuitively, we are looking for a way to monitor $v(t)$ such that:
Requirement 1: the local constraints in the shape of constructed balls are tighter than those of the original framework (Section 3.1)
Requirement 2: the choice of the reference point should be as close as possible to e (due to the establishment of Lemma 2)
since this pair of requirements renders the containment of new constraints in $\bigcup_{i=1}^{n} B_{\frac{e+u_i}{2}}^{\left\| \frac{e-u_i}{2} \right\|}$ more likely. Furthermore, we wish to invent an algorithm that avoids any communication among the sites, unless a threshold crossing is observed.

Since $v(t) = \frac{\sum_{i=1}^{n} w_i u_i^p}{\sum_{i=1}^{n} w_i}$ and $v(t) = \frac{\sum_{i=1}^{n} w_i u_i}{\sum_{i=1}^{n} w_i}$, for any $\mu \in R$ we can express the true global vector as $v(t) = \frac{\sum_{i=1}^{n} w_i(\mu u_i^p + (1-\mu)u_i)}{\sum_{i=1}^{n} w_i}$. So, in order to monitor the current status of the true global vector we may reside to a new convex hull, namely $Conv(\mu u_1^p + (1-\mu)u_1, \dots, \mu u_n^p + (1-\mu)u_i)$. We then find ourselves concerned with identifying a value for μ that may fulfill Requirements 1 and 2.

LEMMA 4. *For any $\frac{1}{2} \leq \mu \leq 1$, when Property 1 holds, tighter local constraints compared to the framework of Section 3.1 are guaranteed, i.e.:* $\|(\mu u_i^p + (1-\mu)u_i) - (\mu e^p + (1-\mu)e)\| \leq \|u_i - e\|$

PROOF.

$$\|u_i^p - e^p\| \leq \|u_i - e\| \overset{\mu \geq 0}{\Leftrightarrow} \|\mu u_i^p - \mu e^p\| \leq \|\mu u_i - \mu e\| \Leftrightarrow$$

$$\|\mu u_i^p - \mu e^p + (1-\mu)(u_i - e) + (\mu-1)(u_i - e)\| \leq \|\mu u_i - \mu e\|$$

By the triangle inequality:

$$\|\mu u_i^p - \mu e^p + (1-\mu)(u_i - e)\| - \|(\mu-1)(u_i - e)\| \leq \|\mu u_i - \mu e\| \Leftrightarrow$$

$$\|\mu u_i^p - \mu e^p + (1-\mu)(u_i - e)\| \leq (2\mu-1)\|u_i - e\|$$

Obviously, $(2\mu-1) \geq 0 \Leftrightarrow \mu \geq \frac{1}{2}$. The balls that are build by the original framework possess a radius of $\frac{\|u_i - e\|}{2}$ and for $\mu \leq 1$:

$$\|\mu u_i^p - \mu e^p + (1-\mu)(u_i - e)\| \leq (2\mu-1)\|u_i - e\| \leq \|u_i - e\|$$

The latter inequality completes the proof. $\quad\square$

The lemma above provides a rough upper, as well as lower, bound to the value of μ such that $\|(\mu u_i^p + (1-\mu)u_i) - (\mu e^p + (1-\mu)e)\| \leq \|u_i - e\|$ in every site. This means that sites construct tighter constraints than the ones they possessed using the original framework.
The Average Model. Lemma 4 shows that setting $\mu = \frac{1}{2}$ meets Requirement 1 and simultaneously provides beforehand some minimum knowledge with respect to the closest we can move $\mu e^p + (1-\mu)e$ towards e for Requirement 2 to be satisfied as well. Based on these, we are able to devise a first simpler alternative to the containment of convex hulls notion, which we term as the *"average*
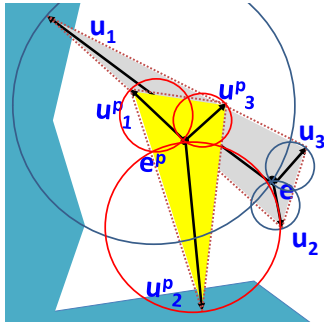
(a) The original and the prediction-based convex hulls cross the threshold. The blue and red balls depict the areas monitored by each site, correspondingly, for the original and the prediction-based frameworks. S_2 violates Property 1 producing a larger prediction deviation (u_2^p) than the corresponding drift vector's (u_2) length.
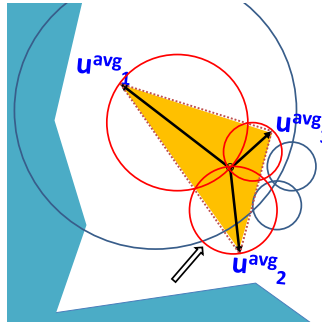
(b) Convex hull and local constraints of the Average Model. Threshold crossing is prevented with stricter local constraints (except for S_2) and increased $\bigcup_{i=1}^{n} B_{\frac{e^p+e+u_i^p+u_i}{4}}^{\|\frac{e^p+e}{4}-\frac{u_i^p+u_i}{4}\|}$ area contained in the constraints of the original bounding algorithm.

Figure 3: The effect of the Average Model Adoption

Figure 4: Loosened Intersection Monitoring. $\max_{i=1..n} B_e^{\|e-u_i\|}$, $\max_{i=1..n} B_{e^p}^{\|e-u_i\|}$ **are produced by** S_1 **which is the one that checks** $\max_{i=1..n} B_e^{\|e-u_i\|} \cap \max_{i=1..n} B_{e^p}^{\|e-u_i\|}$. **No threshold crossing occurs despite that individual convex hulls violate the threshold surface.**

model". The average model monitors $Conv\left(\frac{u_1^p+u_1}{2}, \ldots, \frac{u_n^p+u_n}{2}\right) \subset \bigcup_{i=1}^{n} B_{\frac{e^p+e+u_i^p+u_i}{4}}^{\|\frac{e^p+e}{4}-\frac{u_i^p+u_i}{4}\|}$ by a priori picking a value of $\mu = \frac{1}{2}$.

Figure 3 depicts an example of the Average Model adoption, where both the original and the prediction-based convex hulls cross the threshold surface in different areas (we used three sites to simplify the exposition). In Figure 3(a), notice that for S_2 Property 1 is violated. Despite this fact, as shown in Figure 3(b), the Average Model can still ward off threshold crossing, nearly achieving containment of its spheres in those of the original bounding algorithm.

The Safer Model. We now discuss an alternative model that relaxes Requirement 2. Following a rationale similar to [21], we observe that at any given time, the sites can individually choose the reference point $\mu e^p + (1-\mu)e, \frac{1}{2} \leq \mu \leq 1$ which is farther from the threshold surface and, simultaneously, ensures smaller local constraints. Note that by being far from the threshold surface, a reference point makes the local constraints of any predictor based monitoring model less possible to cause a crossing [21]. This second alternative is termed as the *"safer model"*.

At the first step of the algorithm, every site starts with $\mu_1 = \frac{1}{2}$ and calculates $\mu_1 e^p + (1-\mu_1)e$. In addition, let e_1^* denote the vector lying on the threshold surface and being the closest to $\mu_1 e^p + (1-\mu_1)e$. Every site is capable of individually computing $\|\mu_1 e^p + (1-\mu_1)e - e_1^*\|$ and, thus, determine the distance the first examined reference point yields. To restrain the computational intensiveness of the technique, we define a number of allowed steps θ, such that in every subsequent step $1 \leq j \leq \theta$ the sites employ a value of $\mu_j = \mu_{j-1} + \frac{1}{2\theta}$ until $\mu_\theta = 1$. Eventually, the μ_j value that induces the largest distance is chosen. Notice that using this framework, the sites can reach a consensus regarding μ without any additional communication. This happens due to the fact that the choice of the final μ is based on common criteria related to the threshold surface and the e, e^p vectors that are known to all sites.

6.2 Loosened Intersection Monitoring

So far in this section we have proposed simplistic alternatives that relax the convex hull containment condition that was discussed in Section 5.1. The presented (*average* and *safer*) predictor based monitoring models do manage to avoid any direct communication
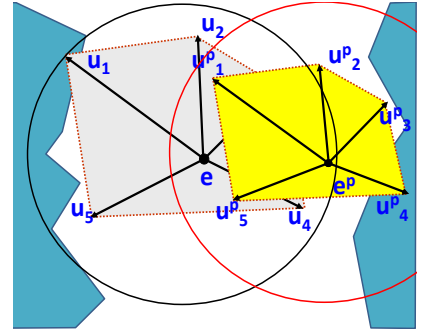
between the sites unless a threshold crossing is detected. Although they do not necessarily abide by Property 2, these models encompass Requirements 1 and 2 (for the average model) and are, thus, in practice likely to substantiate a condition that is hard to check in a distributed manner.

We next aim at inventing a loosened version for the intersection monitoring model of Section 5.2. As before, we wish to come up with a mechanism that avoids any communication between sites unless a threshold crossing happens and simultaneously makes Property 2 very likely. Property 1 is again set as a simple prerequisite, but note that all our algorithms in this section remain correct even if it does not hold, since local constraints still totally cover the monitored area of the input domain. In Section 5.2 we saw that $v(t) \in Conv(u_1, \cdots, u_n) \cap Conv(u_1^p, \cdots, u_n^p)$ while in this section we demonstrated that $v(t)$ also lies in any $Conv\left(\mu u_1^p + (1-\mu)u_1, \cdots, \mu u_n^p + (1-\mu)u_i\right)$ which for $\frac{1}{2} \leq \mu \leq 1$ possesses the desired characteristics formulated in Requirements 1 and 2. The following lemma provides a primitive result on how the intersection monitoring can be achieved using the aforementioned logic. For ease of exposition, we use $Conv_\cap^3$ to denote the triple intersection of these three (original, predicted and weighted) convex hulls, while $Sur(Conv_\cap^3)$
$\equiv \max_{i=1..n} B_e^{\|e-u_i\|} \cap \max_{i=1..n} B_{e^p}^{\|e^p-u_i^p\|} \cap \max_{i=1..n} B_{\mu e^p+(1-\mu)e}^{\|\mu e^p+(1-\mu)e-\mu u_i^p-(1-\mu)u_i\|}$,
where $\max_{i=1..n} B_c^{\|r\|}$ denotes the corresponding (in each maximization term) ball of maximum radius.

LEMMA 5. *For any $\mu \in R$, the area inscribed in $Conv_\cap^3$ is covered by the region induced by $Sur(Conv_\cap^3)$.*

PROOF. Initially notice that:

$$Conv(u_1, \ldots, u_n) \subset \bigcup_{i=1}^{n} B_{\frac{e+u_i}{2}}^{\|\frac{e-u_i}{2}\|} \subset \max_{i=1..n} B_e^{\|e-u_i\|} \qquad (2)$$

$$Conv(u_1^p, \ldots, u_n^p) \subset \bigcup_{i=1}^{n} B_{\frac{e^p+u_i^p}{2}}^{\|\frac{e^p-u_i^p}{2}\|} \subset \max_{i=1..n} B_{e^p}^{\|e^p-u_i^p\|} \qquad (3)$$

$$Conv(\mu u_1^p + (1-\mu)u_1, \ldots, \mu u_n^p + (1-\mu)u_n) \subset$$
$$\bigcup_{i=1}^{n} B_{\frac{\mu e^p+(1-\mu)e+\mu u_i^p+(1-\mu)u_i}{2}}^{\|\frac{\mu e^p+(1-\mu)e-\mu u_i^p-(1-\mu)u_i}{2}\|} \subset \max_{i=1..n} B_{\mu e^p+(1-\mu)e}^{\|\mu e^p+(1-\mu)e-\mu u_i^p-(1-\mu)u_i\|} \qquad (4)$$

So each time the maximum balls cover the corresponding convex hulls. We want to prove that the intersection of the latter balls also covers the intersection of the convex hulls. The proof will be derived by contradiction.

Suppose that a vector $h \in Conv_\cap^3$ exists. Now assume that the vector h does not lie in at least one of $\max\limits_{i=1..n} B_e^{\|e-u_i\|}$, $\max\limits_{i=1..n} B_{e^p}^{\|e^p-u_i^p\|}$, or $\max\limits_{i=1..n} B_{\mu e^p+(1-\mu)e}^{\|(\mu e^p+(1-\mu)e)-(\mu u_i^p+(1-\mu)u_i)\|}$. However, this would violate at least one of the Propositions 2,3,4, which is a contradiction. This concludes the proof. $\square$

Reckon, however, that the ascertainment of Lemma 5 cannot be tracked in a distributed manner. This happens because the site that determines each maximum ball may be different. Should Property 1 and, thus, (for $\frac{1}{2} \leq \mu \leq 1$) Lemma 4 hold, what sites actually need to perform so that they can distributively track $Conv_\cap^3$ is to use $B_e^{\|e-u_i\|}$, $B_{e^p}^{\|e-u_i\|}$ and $B_{\mu e^p+(1-\mu)e}^{\|e-u_i\|}$. To understand this, please observe that if both $\|e^p - u_i^p\|$, $\|(\mu e^p + (1-\mu)e) - \mu u_i^p - 1 - \mu)u_i\| \leq \|e-u_i\|$ (due to Property 1 and Lemma 4, respectively), it is evident that: $Sur(Conv_\cap^3) \subset \max\limits_{i=1..n} B_e^{\|e-u_i\|} \cap \max\limits_{i=1..n} B_{e^p}^{\|e-u_i\|} \cap \max\limits_{i=1..n} B_{\mu e^p+(1-\mu)e}^{\|e-u_i\|}$.

Hence, the site that possesses the maximum $\|e-u_i\|$ will check whether the intersection of its locally constructed balls crosses the threshold. Provided that the intersection of the local balls does not cross the threshold at any site (and, thus, at the site with the maximum $\|e-u_i\|$ as well), synchronization can safely be avoided. At this point, we would be interested in identifying proper values for μ that refine the range ($\frac{1}{2} \leq \mu \leq 1$) established in Lemma 4. Nonetheless, the following corollary shows that if we have to employ $\|e-u_i\|$ as the radius of the balls, $\max\limits_{i=1..n} B_{\mu e^p+(1-\mu)e}^{\|e-u_i\|}$ does not reduce the volume of the intersection.

COROLLARY 1. *When Property 1 holds and for any* $0 \leq \mu \leq 1$, $\max\limits_{i=1..n} B_{\mu e^p+(1-\mu)e}^{\|e-u_i\|}$ *does not reduce the volume of the region induced by* $\max\limits_{i=1..n} B_e^{\|e-u_i\|} \cap \max\limits_{i=1..n} B_{e^p}^{\|e-u_i\|}$.

PROOF. Omitted due to space considerations. $\square$

Corollary 1 is true for any $0 \leq \mu \leq 1$, but note that in order to ensure $\|e^p - u_i^p\| \leq \|e-u_i\|$ and $\|(\mu e^p + (1-\mu)e) - \mu u_i^p - (1-\mu)u_i\| \leq \|e-u_i\|$, we had already assumed that $\frac{1}{2} \leq \mu \leq 1$. Hence it suffices to check whether the pair $B_e^{\|e-u_i\|} \cap B_{e^p}^{\|e-u_i\|}$ crosses the threshold in at least one site [2]. Figure 4 provides an exemplary application of the intersection monitoring procedure described so far, where $\max\limits_{i=1..n} B_e^{\|e-u_i\|}$, $\max\limits_{i=1..n} B_{e^p}^{\|e-u_i\|}$ are produced by S_1.

On the other hand, following an intuition similar to the one utilized in the average model, an alternative is to track $Conv(u_1^p, \cdots, u_n^p) \cap Conv(\frac{u_1^p+u_1}{2}, \cdots, \frac{u_n^p+u_n}{2})$ instead. In other words, this time each site S_i needs to individually construct two balls using e^p and $\frac{e^p+e}{2}$ as centers and $M = max\{\|e^p - u_i^p\|, \|\frac{e^p+e}{2} - \frac{u_i^p+u_i}{2}\|\}$ as the common radius (please note that M refers to the maximum of the pair of local radii). Subsequently, a synchronization is caused when at least one S_i detects that the locally constructed intersection crosses the threshold.

We conclude our study by showing the condition which makes the latter intersection tracking preferable as it results in smaller local constraints compared to $\max\limits_{i=1..n} B_e^{\|e-u_i\|} \cap \max\limits_{i=1..n} B_{e^p}^{\|e-u_i\|}$.

---

[2]Even if the site that determines the maximum radius finds that Property 1 does not hold, Corollary 1 is still valid upon replacing $\|e-u_i\|$ with $\|e^p - u_i^p\|$.

PROPOSITION 1. *When* $\max\limits_{i=1..n} B_e^{\|e-u_i\|} \supseteq \max\limits_{i=1..n} B_{\frac{e^p+e}{2}}^{M}$, *then:*

$$\max\limits_{i=1..n} B_{e^p}^{M} \cap \max\limits_{i=1..n} B_{\frac{e^p+e}{2}}^{M} \subseteq \max\limits_{i=1..n} B_e^{\|e-u_i\|} \cap \max\limits_{i=1..n} B_{e^p}^{\|e-u_i\|}$$

PROOF. Omitted due to space considerations $\square$

6.3 Choosing Amongst Alternatives

So far, we investigated a number of simpler alternatives that loosen the strong monitoring frameworks of Section 5, i.e., the convex hull containment as well as the intersection monitoring framework. We based our analysis on Property 1 as an intuitive assumption also employed in past studies [5, 2, 4] and evolved it to practical tracking mechanisms together with appropriate speculative analysis. Nonetheless, upon relaxing the monitoring conditions we also relaxed their conformity to Property 2, i.e., the prerequisite for strong predictor-based monitoring models. Since the coordinator is supposed to a priori dictate the predictor-based tracking alternative that should be uniformly utilized by sites at least until the next synchronization, we need to provide a decision making mechanism that enables it choose among the available options and adjust its decisions on their anticipated performance with respect to communication savings.

The available tracking options that do not belong (excluding the trivial choice of the original framework) to the strong predictor-based monitoring models' class include:

- Monitoring of $Conv(u_1, \ldots u_n)$ as in Section 3.1
- Monitoring of $Conv(u_1^p, \ldots, u_n^p)$ as in Section 4
- Adoption of the average model
- Adoption of the safer model
- Tracking of $\max\limits_{i=1..n} B_e^{\|e-u_i\|} \cap \max\limits_{i=1..n} B_{e^p}^{\|e-u_i\|}$
- Tracking of $\max\limits_{i=1..n} B_{e^p}^{M} \cap \max\limits_{i=1..n} B_{\frac{e^p+e}{2}}^{M}$

In order to provide an appropriate decision making mechanism, we require that sites keep up monitoring all the six options mentioned above. This monitoring will take place **only** for models that would not result in **any** local transmission since the last synchronization (i.e., we stop monitoring an alternative model for which we detect that a transmission would have been caused). Notice that one model has been chosen as the main model after the last synchronization. Thus, for each of the 6 alternatives, the sites maintain 6 bits, where the i-th bit is set iff the corresponding monitoring mode would have resulted in at least one transmission since the last synchronization. A synchronization can still be caused only by the main model. Upon a synchronization, however, together with $v_i(t)$ and the velocity vector in the case of the velocity acceleration model choice, sites attach 5 bits (they do not need to send a bit for the current model being used) on their messages. Please note that the following facts hold in our adaptive algorithm:

- No site had a violation using the current model in a previous time instance (since the previous synchronization).

- An alternative model that has its corresponding bit to 1 in **any** of the sites would not have been better than the model currently being used, since it would have resulted in a transmission in a prior (or the current) time instance.

- Based on the above observation, we decide to switch to an alternative model only if the corresponding bits for this model were equal to 0 in **all** the sites.

In case of multiple alternatives with unset bits, a random choice among such alternatives is performed.

7. EVALUATION RESULTS

In order to evaluate our algorithms we developed a simulation environment in Java. We utilized two real data sets to derive data stream tuples arriving at every site in the network. "Corpus", consists of 804,414 records present in the Reuters Corpus (RCV1-v2) [14] collection. Each record corresponds to a news story to which a list of terms (features) and appropriate categorization have been attributed. As in [21, 20] we focused on the following features: *Bosnia, Ipo, Febru* while monitoring their coexistence with the *CCAT* (the CORPORATE / INDUSTRIAL) category. Aiming at identifying the relevance of these features to the CCAT category at any given time, we monitored two different functions involving the Chi-Square(χ^2) and Mutual Information(*MI*) score. We utilized the Corpus data set in order to test our techniques using the Cash Register streaming paradigm i.e. taking into consideration the whole history of the tuples arriving at the various sites. In any given timestamp, after the receipt of a new tuple each site forms a vector which consists of four dimensions for the χ^2 and three dimensions for the *MI* case. These vectors have one of their positions set, while the rest remain zero. In particular, for both the functions the first position of the vector is set if the term and the category co-occur, the second if the term occurs without the CCAT category, the third in case CCAT is present without the term, while the fourth (only for χ^2 score) if neither of them appeared in the incoming tuple.

Due to the nature of the incoming (binary) vectors and the utilized Cash Register paradigm the previously described environment may be considered moderate to change and be thought of as easily predictable by our techniques. In order to test our algorithms in more dynamic conditions we utilized one more data set. The "Weather" data set includes Solar Irradiance, Wind Speed and Wind Peak measurements from the station in the University of Washington and for the year 2002 [9], where each file incorporates 523439 records of measurements. We used the Weather data sets so as to monitor the Variance (Var) and the Signal to Noise Ratio (StN) functions. We utilized Var since it has already been used within the geometric monitoring framework [19]. In addition, the StN function equals the ratio between the mean and the standard deviation ($\frac{\mu}{\sigma}$) in a given window of measurements and can, thus, be applicable to globally quantify the noise present in the measurements.

In each experiment we first measure the number of messages transmitted in the network across different thresholds for a network configuration consisting of 10 sites. We then use the middle case threshold and plot the number of transmitted messages when increasing the scale of the distributed environment (the number of sites). We denote the performance of the original bounding algorithm (Section 3.1) by "Model 0", while "Model 1" refers to the mere application and monitoring of the prediction-based bounding algorithm (Section 4). Eventually, "CAA" shows the performance of the Choosing Amongst Alternatives framework that was introduced in Section 6.3. Moreover, for each of the lines in the graphs, we enclosed the chosen predictor using *LG* to denote the Linear Growth predictor and *VA − W* so as to declare a Velocity / Acceleration predictor with a window of *W* measurements [3].

7.1 Corpus Data Set - Cash Register Paradigm

We begin our study by examining the performance of our techniques in the Corpus data set on par with the Cash Register paradigm

[3]We focus on comparing the performance of our prediction-based geometric monitoring techniques against [18, 20] (Model 0), since we expect our prediction-based methods to give similar benefits when operating over the ellipsoidal bounding regions of [21] to those seen in our current study using spherical constraints as in [18, 20].

adoption. Figure 5 depicts the performance of Model 0 and of the CAA approach when using the *LG* and the *VA* predictors. Since almost 6000 documents are received within a period of a month [21], we choose a $W = 200$ window for the VA predictor which is expected to be roughly the amount of news stories received daily. Each column of the figure corresponds to the case of the terms "Bosnia", "Ipo" and "Febru", respectively.

Sensitivity to Threshold - Chi Square. As shown in the first column of Figure 5, where the χ^2 function for the term "Bosnia" is monitored, Model 0 appears to always be about 2 and 1.85 times worse in terms of the number of transmitted messages when compared to the CAA(LG) and CAA(VA-200) approaches, respectively, for different threshold values (Fig. 5(a)) using 10 sites.

Sensitivity to Threshold - Mutual Information (MI). Moving to the second and third columns of Figure 5(a) we investigate the cases of the "Ipo" as well as "Febru" terms, monitoring the *MI* function across different threshold values for a 10 site configuration (note that *MI* is calculated as a logarithm, therefore the negative threshold values in that axis). In these graphs the peak that occurs at 0.4 and 0 for the "Ipo" and "Febru" involves an accumulation of synchronizations around the average value the *MI* function possesses along the run. We again observe that CAA(LG) performs 1.75-2.1 times better than the Model 0 case for both monitored terms. Despite the fact that CAA(VA-200) is proved slightly worse compared to CAA(LG), it is still able to better amend the peak that occurs in "Febru" monitoring for a 0 threshold.

Sensitivity to Number of Sites. Eventually, switching to Figure 5(b), for the "Bosnia" term (left column) the relative benefits remain almost the same across all network scales. For the "Ipo" term monitoring (middle column) we observe that Model 0 is steadily more than 1.8 times worse than the CAA(LG) choice across different scales. CAA(VA-200) performs worse than the CAA(LG) case for network configurations up to 80 sites. Nonetheless, the introduction of additional sites (along with their respective substreams) in 90, 100 site cases, causes the *MI* function to always lie below the posed zero threshold since the "Ipo" term becomes more rare. This fact is perfectly read by the CAA(VA-200) approach, the transmitted messages of which approach zero. A similar behavior appears early in the third column of Figure 5(b) for the Febru case where the introduction of more than 10 sites causes *MI* to be negative, which is again accurately pinpointed by the CAA(VA-200) monitoring model reaching savings of 3 orders of magnitude size.

In the previously presented graphs we omitted the lines for the mere application of Model 1 to keep the diagram readable, since Model 1 shows almost identical (actually CAA can occasionally save a few tens of extra messages) behavior with its corresponding CAA applications. CAA possesses the ability to recognize the utility of Model 1 (the mere application of predictors as described in Section 4) in this setting and encompass it throughout its operation. On the other hand, this fact exhibits the ability of Model 1 to provide an efficient solution in environments where v_i values evolve relatively slowly. Nonetheless, as we will shortly present, this is not always true in scenarios where more dynamic updates occur.

7.2 Weather Data - Sliding Window Paradigm

We proceed to the sliding window operation, using the Weather data set and monitoring the Var and StN functions. Please note that in all the graphs presented in the current subsection the Linear Growth predictor is not applicable since it assumes that local vectors (v_i) uniformly evolve by a time dependent factor (Section 3.2), which is obviously unrealistic for the physical measurements in the Weather data and the sliding window application scenario. We thus compare the performance of Model 0, Model 1(VA-W) and

(a) Cash Register Varying Threshold
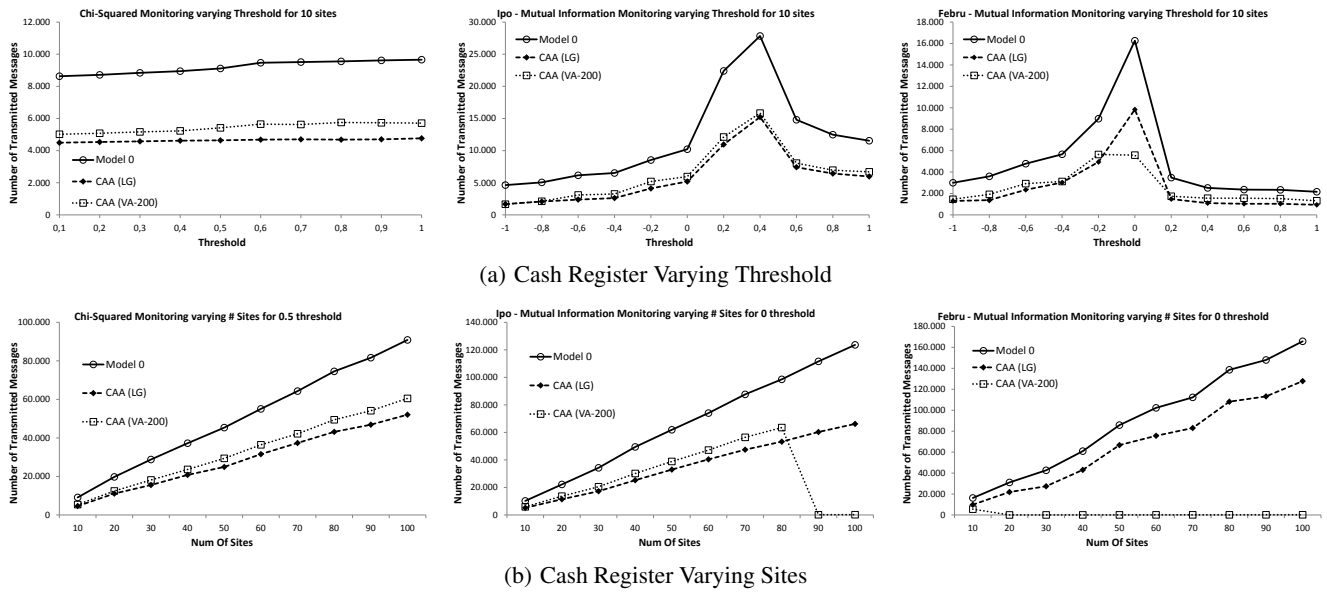

(b) Cash Register Varying Sites

Figure 5: Corpus Data Set: Performance of our Techniques in the Cash Register Streaming Paradigm

CAA(VA-W) cases in our study. For the CAA(VA-W) monitoring model we again choose the window based on natural time units' division. We uniformly utilize a prediction window $W = 10$ which corresponds to the latest minutes of received observations, except for the Wind Peak data where $W = 50$ was chosen to adjust predictions to the expected frequency of the peaks in the wind blasts. For each data set-function pair, the default value of the sliding window size, over which the corresponding function is computed, is 200 measurements. However, we also perform a sensitivity analysis on this parameter as well.

Variance Monitoring. Figure 6(a) plots the performance of the techniques in the case of Var monitoring in the Solar Irradiance Data. In the first column of the figure we observe that the cost of Model 0 ranges between 11 and 600 times larger than the cost yielded by CAA(VA-10) monitoring model, while CAA(VA-10) ensures up to 500 times lower cost even when compared with Model 1 across different thresholds. A case of particular interest shows up for a threshold of 30000. There, Model 1 shows a peak in the number of transmitted messages which are higher even when compared to Model 0. This happens due to the existence of specific sites whose drift vectors approach the threshold surface as noted in Figure 2. Obviously, increasing the threshold to 40000 alters the threshold surface and thus hinders the same sites to cause threshold violations. Nonetheless, CAA(VA-10) maintains low transmission cost due to the loosened intersection monitoring capabilities (Section 6.2) that it embodies. We will revisit this issue in the next subsection where we look into the operational details of the CAA monitoring model. In the meantime we note that the same applies for the second column of Figure 6(a) where increasing the scale of the network results in CAA(VA-10) savings that reach a number of 30 times compared to Model 0 and they become even larger when compared to Model 1. Eventually, the third column of the same figure, shows the resilience of our techniques when altering the employed size of observations encapsulated in the sliding window for 10 sites. CAA(VA-10) shows similar behavior when enlarging the window. Model 1 yields more synchronizations for a window of 200 observations since enlarging the window causes the variance values within it to increase and thus some sites approach the posed

threshold of 50000. The lack of the alternative mechanisms that are incorporated in CAA leads sites merely utilizing Model 1 to threshold crossings. Finally, Model 0 exhibits high sensitivity to the number of values that local vectors (v_i) are built upon.

Figure 6(b) presents corresponding results for Var monitoring in the Wind Speed data set. Model 1 is slightly worse (almost 1%) in terms of transmitted messages compared to CAA(VA-10) when varying the threshold (first column in the figure) and across different network scales (second column), yet both result in savings ranging between 3 and 13 times compared to the message cost of Model 0. Furthermore, in the third column of Figure 6(b) we observe that both CAA(VA-10) and Model 1 remain resilient to altering the sliding window size ensuring significant benefits when compared to Model 0. Notice that for a window of 100 observations, Model 1 performs better than CAA(VA-10). Recall that CAA resolves ties in the choice of the monitoring mechanism (Section 6.3) by picking a random model among those which did not cause a threshold crossing. Thus, when a particular model is always the appropriate choice, the adaptive CAA algorithm may sometimes end up transmitting slightly more messages. The results are similar for the Wind Peak data set which we omit due to space limitations.

Signal to Noise Monitoring. In our next experiment we utilized the same motif for analyzing the performance of our techniques in monitoring the StN function. We begin our discussion with the Solar Irradiance data set in Figure 7(a). CAA(VA-10) performs up to 3 times better than Model 0 when varying the threshold for 10 sites (first column in the figure) and up to 5 times across network configurations of 10-100 sites (second column). Model 1 is again the worst choice as it yields 2-5 times higher cost compared to Model 0 across different thresholds and appears over 2 times worse than Model 0 for different scales. In the third column of Figure 7(a), it is evident that CAA(VA-10) again remains mostly unaffected to different window sizes, while Model 1 exhibits a wavy behavior depending on the accuracy of the employed VA-10 predictor.

We then analyze the performance of the Wind Peak Data in StN monitoring (Figure 7(b)) (the Wind Speed data had similar behavior). Model 1 and CAA possess similar performance across different thresholds with savings ranging between 4 and 85 times
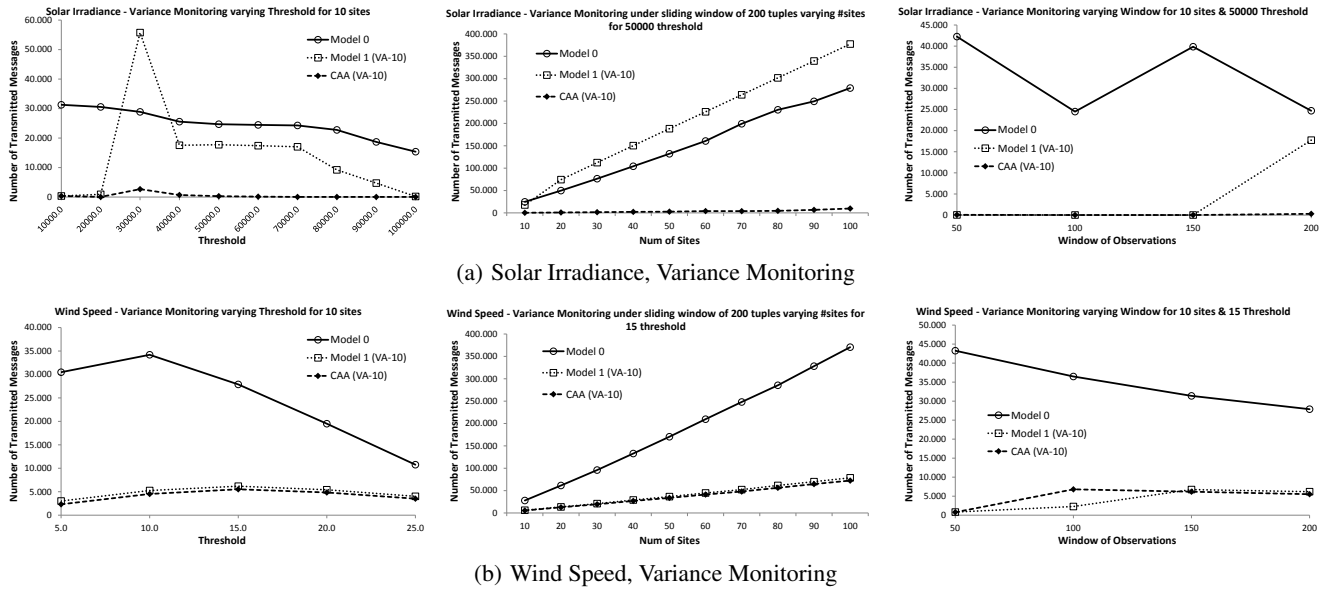
(a) Solar Irradiance, Variance Monitoring

(b) Wind Speed, Variance Monitoring

Figure 6: Weather Data Set: Performance of our Techniques in the Sliding Window Streaming Paradigm for Variance Monitoring

Table 3: Case study: Solar-Var Vs Threshold Monitoring

Threshold	Model 0	Model 1	Average	Safer	Intersection 1	Intersection 2
10000	0	15	0	3	0	0
30000	105	0	0	0	4	25
50000	7	0	0	0	5	3
70000	0	0	1	0	0	1
90000	0	0	1	0	0	1

Table 4: Case study: Wind Peak-StN Vs # Sites Monitoring

# Sites	Model 0	Model 1	Average	Safer	Intersection 1	Intersection 2
10	35	16	25	35	1	3
40	8	6	12	19	0	1
50	13	10	7	17	0	1
80	12	14	9	14	0	1
90	9	9	12	21	0	1

compared to the cost of Model 0. The same holds in large part when varying the network scale (middle column in Fig 7(b)) where savings reach a factor of 5. An exception occurs for 40 and 50 site configuration cases. This is another occasion where site predictors lie close to the threshold surface for the given threshold of 0.5 and CAA manages to achieve increased savings due to the intersection monitoring capacity. As more sites are added in the subsequent steps (60-100 site configurations) the predicted estimate's (e^p) position is affected and thus the sites that were previously causing synchronizations (despite their restricted local constraints - balls) were ousted from the threshold surface, stabilizing the cost of Model 1. Eventually, as with the previously examined functions-data set pairs, the CAA monitoring model is not sensitive to changes in the window size (third column of Fig 7(b)) while Model 0 and Model 1 exhibit opposite trends upon enlarging it. Overall, Model 0's cost is 5 to 35 times the transmission cost of CAA, while savings against Model 1 range between 4 to 9 times across different window sizes.

7.3 CAA Operational Insights

We are now providing additional details regarding the choices that CAA makes throughout its operation to investigate the stem of its benefits. Since it is hard to present analytic statistics of alternative models' usage for every single case of the previously discussed graphs, we focus on two situations where Model 1 exhibits possibly unexpected peaks in the number of messages and examine the tools that CAA utilizes to avoid similarly high message exchange.

The first of the aforementioned cases regards the Solar Irradiance under Var monitoring against different thresholds and for 10 sites(left figure of Fig 6(a)). Table 3 shows the CAA choices for

different thresholds. Intersection1 refers to monitoring the intersection between the original and the predicted convex hull, while Intersection2 refers to monitoring the intersection between the average convex hull and the predicted one. We point out that for threshold >10000 (where it exhibits low costs) Model 1 is never employed by CAA. For the threshold 30000 case, Model 0 appears as the most frequent choice but it is only used during the first synchronizations until predictors are stabilized around the threshold surface (if Model 0 was continuously picked, CAA would have had similar cost to Model 0). Afterwards, the loosened intersection framework is chosen which safely leads the monitoring procedure to the decrement of the transmission cost as shown in Fig 6(a).

The second case we distinguished during our discussion was the peak that occurs when monitoring the Wind Peak data under the StN function for network configurations of different scale (middle figure of Fig. 7(b)). As Table 4 shows, for 10 sites the savings CAA provides are mostly attributed to the average and safer model usage, while for 40, 50 and more sites, after a few synchronizations, the single time that Intersection2 is employed by CAA hinders communication for a considerable amount of time.

8. CONCLUSIONS

In this paper, we presented a thorough study regarding prediction models' adoption within the geometric monitoring setting. After identifying the peculiarities exhibited by predictors upon their implementation in the aforementioned environment, we developed a solid theoretic framework composed of sufficient conditions rendering predictors capable of refraining the communication burden. We proposed algorithms incorporating those conditions and ex-
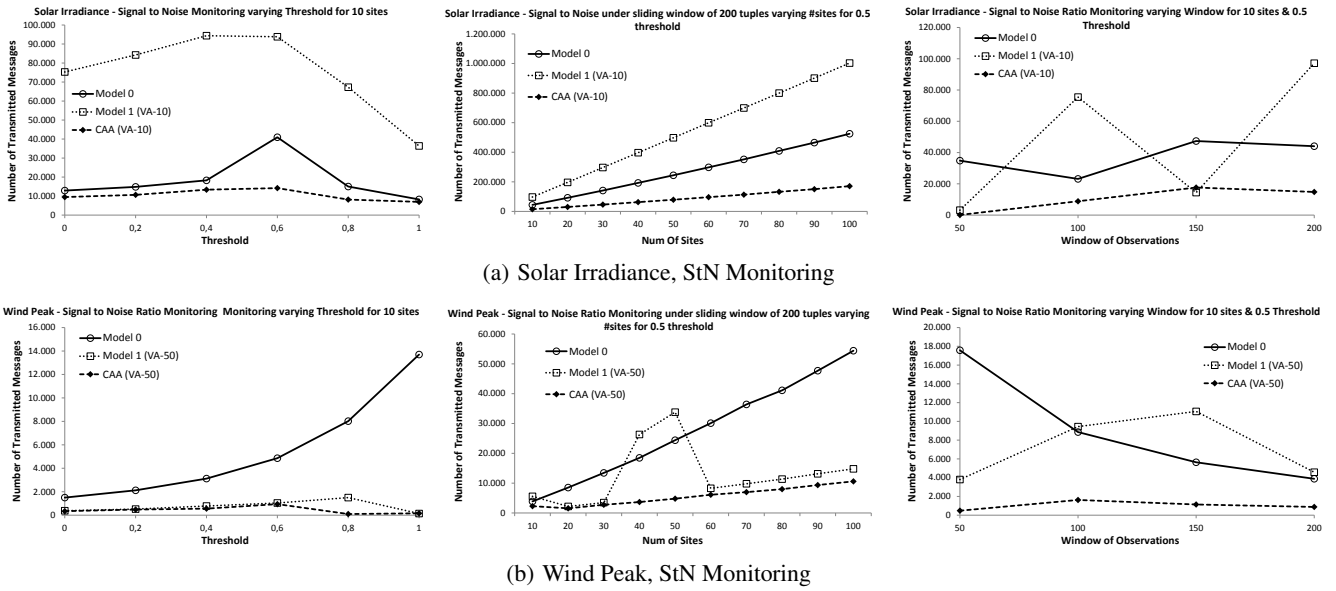
(a) Solar Irradiance, StN Monitoring

(b) Wind Peak, StN Monitoring

Figure 7: Weather Data Set: Performance of our Techniques in the Sliding Window Streaming Paradigm for StN Monitoring

panded on relaxed versions of them along with extensive theoretical analysis on their expected benefits. Our ongoing efforts in this area explore the choice of optimal reference points, as in [21], that could perhaps enable "looser" conditions of strict containment.

9. REFERENCES

[1] B. Babcock and C. Olston. Distributed top-k monitoring. In *SIGMOD*, 2003.

[2] G. Cormode and M. Garofalakis. Sketching streams through the net: Distributed approximate query tracking. In *VLDB*, 2005.

[3] G. Cormode and M. Garofalakis. Streaming in a connected world: querying and tracking distributed data streams. In *SIGMOD*, 2007.

[4] G. Cormode and M. Garofalakis. Approximate continuous querying over distributed streams. *ACM Transactions on Database Systems*, 33(2), 2008.

[5] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Holistic aggregates in a networked world: distributed tracking of approximate quantiles. In *SIGMOD*, 2005.

[6] G. Cormode, S. Muthukrishnan, and K. Yi. Algorithms for distributed functional monitoring. *ACM Trans. Algorithms*, 7:21:1–21:20, 2011.

[7] G. Cormode, S. Muthukrishnan, and W. Zhuang. Conquering the divide: Continuous clustering of distributed data streams. In *ICDE*, 2007.

[8] A. Das, S. Ganguly, M. Garofalakis, and R. Rastogi. Distributed set-expression cardinality estimation. In *VLDB*, 2004.

[9] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Compressing Historical Information in Sensor Networks. In *ACM SIGMOD*, 2004.

[10] L. Huang, M. Garofalakis, J. Hellerstein, A. Joseph, and N. Taft. Toward sophisticated detection with distributed triggers. In *MineNet*, 2006.

[11] L. Huang, X. Nguyen, M. Garofalakis, and J. M. Hellerstein.

[12] A. Jain, J. M. Hellestein, S. Ratnasamy, and D. Wetherall. A wakeup call for internet monitoring systems: The case for distributed triggers. In *HotNets*, 2004.

[13] R. Keralapura, G. Cormode, and J. Ramamirtham. Communication-efficient distributed monitoring of thresholded counts. In *SIGMOD*, 2006.

[14] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[15] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30:122–173, March 2005.

[16] C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In *SIGMOD*, 2003.

[17] G. Sagy, D. Keren, I. Sharfman, and A. Schuster. Distributed threshold querying of general functions by a difference of monotonic representation. *PVLDB Endow.*, 4:46–57, 2010.

[18] I. Sharfman, A. Schuster, and D. Keren. A geometric approach to monitoring threshold functions over distributed data streams. In *SIGMOD*, 2006.

[19] I. Sharfman, A. Schuster, and D. Keren. Aggregate threshold queries in sensor networks. In *IPDPS*, 2007.

[20] I. Sharfman, A. Schuster, and D. Keren. A geometric approach to monitoring threshold functions over distributed data streams. *ACM Transactions on Database Systems*, 32(4), 2007.

[21] I. Sharfman, A. Schuster, and D. Keren. Shape sensitive geometric monitoring. In *PODS*, 2008.

[22] K. Yi and Q. Zhang. Optimal tracking of distributed heavy hitters and quantiles. In *PODS*, 2009.

[23] Q. Zhang, J. Liu, and W. Wang. Approximate clustering on distributed data streams. In *ICDE*, 2008.

Communication-efficient online detection of network-wide anomalies. In *INFOCOM*, 2007.