# Towards Prediction with Partial Data in Sensor-based Big Data Applications

Saima Aman
Department of Computer Science
University of Southern California
Los Angeles, CA
saman@usc.edu

Charalampos Chelmis, Viktor Prasanna
Department of Electrical Engineering
University of Southern California
Los Angeles, CA
{chelmis, prasanna}@usc.edu

***Abstract*—Many emerging big data applications such as in smart electric grids, transportation, avionics, manufacturing, and remote medical and environment monitoring involve sensors for tracking, monitoring, and control. These sensors are generally located at geographically dispersed locations and expected to periodically send back acquired information to centrally located nodes or processing centers. In many cases, the data from sensors is not available at central nodes at a frequency that is required for fast and real-time modeling and decision-making. For example, while many of these sensors are capable of collecting information at a high speed, logging data every minute or so, the physical limitations, specially latency, of the transmission networks limit the frequency at which data from sensors can be transmitted back to the central nodes. Also, consumers may limit frequent transmission of information from sensors located at their premises for security and privacy concerns. Finally, the data may not reach the central nodes due to faults in the sensors or transmission systems. All these scenarios raise the issue of *data veracity* in big data applications. While volume, variety, and velocity aspects of big data have been the focus of much recent research, veracity has received less attention. In this paper, we propose a novel solution to the problem of making short term predictions (up to a few hours ahead) in absence of real-time data from sensors. A key implication of our work is that by using real-time data from only a small subset of *influential* sensors, we are able to make predictions for *all* sensors. We thus reduce unnecessary transmissions from sensors and provide a practical solution to data veracity in many sensor based big data applications. We use real-world electricity consumption data from smart meters to empirically demonstrate the usefulness of our method.**

***Keywords*—*data veracity, short-term prediction, prediction model, smart grid***

## I. Introduction

Low cost wireless sensors are increasingly being deployed in large numbers for performing tracking, monitoring, and control in emerging big data applications such as in smart electric grids, transportation, avionics, manufacturing, and remote medical and environment monitoring. These sensors are generally located at geographically dispersed locations and expected to periodically send back acquired information to centrally located nodes or processing centers [12] via wireless links and the Internet [11], [32]. Examples of such sensors include climate sensors for monitoring features such as temperature, solar radiance, and green-house gas measurements [21]; smart meters for measuring energy consumption [32], [22]; loop detectors installed under pavements for recording traffic [27]; and meters on wind turbines that record wind speed and turbines' power output [10].

Due to several factors, data from all sensors is not available at the central nodes in real-time or at a frequency that is required for fast and real-time modeling and decision-making. For example, while many of these sensors are capable of collecting information at a high speed, logging data every minute or less, *physical limitations* of existing transmission networks, such as latency, bandwidth and high energy consumption [12] are key factors that limit the frequency at which data from sensors can be transmitted back to the central nodes [7]. Thus, sensors either transmit a quantized version of the measurements [12] or collect high resolution data locally but transmit information periodically in batches one or more times a day. For instance, wind turbines record data every few seconds, but transmit data every five minutes to far-off research centers for use in forecasting algorithms [10].

Another factor responsible for non-availability of real-time data at the central nodes is that consumers may opt-out or limit frequent transmission of information from sensors located at their premises for *security and privacy* concerns [23]. For instance, fine-grained electricity consumption data collected through smart meters can be used to infer activities of the consumers and also indicate the presence or absence of dwellers in the consumer premises [24]. Furthermore, sensor data may not reach the central nodes due to faults or outages, and unreliability or shadow fading of transmission links [11]. All these situations reflect the ***partial data problem***, where only partial data from sensors is available in real-time, and complete high resolution data is available only periodically, generally one or more times a day.

Partial data raises the issue of data *veracity* in sensor based big data applications and questions the reliability of models running at central nodes that assume availability of high resolution data in real time. Veracity is closely tied with the other 3 Vs of big data, that is, volume, velocity and variety [20] that respectively describe the increasing size of data, the increasing rate of data generation, and the increasing range of data types used. While these 3 Vs characterize the quantitative aspect of big data, *veracity* characterizes the qualitative aspect of data. Without addressing veracity, big data solutions risk degradation in performance and inaccurate interpretation of generated insights. A possible approach - as discussed in this paper - is to develop creative solutions using data from a small subset of sensors selected on the basis of some heuristics or learning methods, while minimizing information loss resulting from leaving out data from remaining sensors. The intuition behind this approach is that in many cases, sensors may be located spatially close to each other and thus likely to be correlated, or they may measure similar activities driven by similar schedules such as those on an academic campus or traffic on high density roads. If this information can be leveraged, it will obviate the need for real-time transmission from all sensors to the central nodes, and thereby reduce the load on the transmission network. Also, it would make it simpler to add new sensors without straining the network.

In this paper we address the partial data problem in context of smart electricity grids. In smart grid, high *volume* electricity consumption data is collected by smart meters at consumer premises and securely transmitted back to the electric utility over wireless or broadband networks [32], where they are processed for high-level use cases and applications such as planning, customer education, and demand response [3]. A *variety* of other data sources such as weather and daily schedules that may indirectly indicate power consumption [6] are also utilized in smart grid (Figure 1). Transforming the traditional power grid into a smart grid requires learning algorithms that can model data being generated at high *velocity* to predict electricity consumption and use that to initiate curtailment programs ahead of time by the utility to avoid potential supply-demand mismatch [2]. Here we witness partial data problem when data from smart meters is only partially available in real-time. To address this, we propose a novel two-stage solution: First, we learn the dependencies among time series of different smart meters on a similar day in recent past. Then, we use data from a small subset of smart meters which are found to have high *influence* on others to make predictions for all meters. We show that this technique results in only $\sim 0.5\%$ increase in prediction error, while using real-time data from only $\sim 7\%$ of smart meters (Figure 8(b)). Our main contributions are:

1) We propose to leverage dependencies among time series sensor data for making real-time predictions with *partial data*. While time series dependencies have been used previously (Section II), the novelty of our work is in extending the notion of dependencies to discover *influential* sensors and using real-time data only from them to do predictions for all sensors.

2) Using real-world smart grid data, we empirically demonstrate that by trading a fraction of prediction accuracy, we are able to do real time prediction for several hours ahead using real-time data from only a small subset of smart meters.

The rest of the paper is organized as follows. We begin with reviewing related work in Section II. We provide definitions of important terms used in the paper as well as formal problem definition in Section III. Our proposed method is described in Section IV and the experimental results are presented in Section V. We conclude the paper in Section VI.

## II. Related Work

Within big data, research on *veracity* broadly deals with data quality and trustworthiness. Ensuring data quality for big data applications requires detecting and repairing erroneous data in a scalable and timely manner [29]. Data generated from sensors in many big data applications is particularly susceptible to quality issues due to discretization in continuous data, misalignment in data due to out-of-sync sensor clocks, faults, anomalies, and network delays [22]. Trustworthiness depends on factors such as data origin, collection and processing, and trusted infrastructure [14]. Another aspect of veracity - which is also the focus of this paper - is associated with the *availability* and *timeliness* of data [14]. It is an important consideration for many big data algorithms that assume ideal scenarios with all required data readily available.

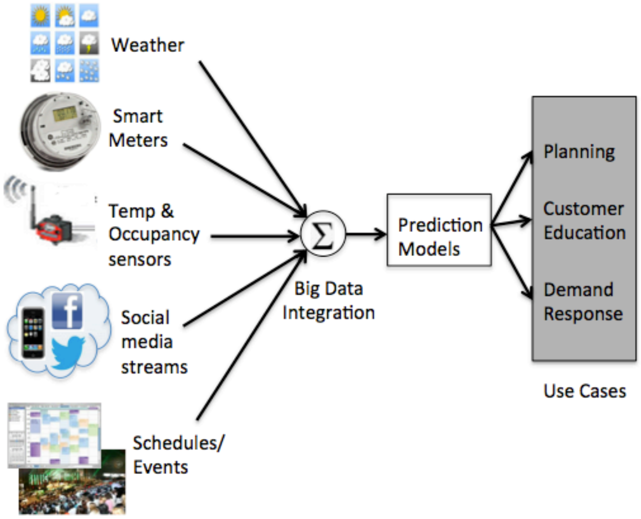


Fig. 1. An illustration of how big data in Smart Grid is utilized for prediction for different use cases.

Generic time-series prediction methods such as Auto-Regressive Integrated Moving Average (ARIMA) [8] and Auto-Regressive Trees (ART) [25] use observations made in recent past to make predictions for short-term future. Existing works assume such past observations to be readily available in real-time. However, as mentioned in Section I, this assumption does not hold true for many sensor-based big data applications because large volumes of data cannot be transmitted over the network in real-time and only partial data is available in real time. The solutions proposed to address this problem can be categorized into two types: 1) In the first type, the approach is to reduce the volume of transmitted data by techniques such as data compression [22], [28], data aggregation [17], and model-driven data acquisition [15]. The need to develop communication efficient algorithms is also highlighted in [30], where only network volume sub-linear in the input size is considered feasible for big data applications. 2) In the second type, attempts are made to estimate missing real-time data by techniques such as interpolation based on regression [18], or through transient time models that use differential equations to model system behavior [13]. Main challenge with these methods is that estimates depend on the accuracy of models and interpolation errors get propagated to subsequent analysis and decision-making steps. Another method for estimation is using spectral analysis of time series, though it is a more complex and involved process that is suitable only for periodic time series [5]. We use a different approach where instead of trying to estimate the missing real-time data, we try to do predictions using partial real time data by learning dependencies among time series originating from different sensors.

Different approaches have been proposed to learn dependencies among time series data; the more popular among them are based on cross-correlations [8] and Granger Causality [16]. The latter is used to determine if the past values of a series help in predicting future values of another series. It has gained popularity in many domains such as climatology, economics, and biological sciences due to its simplicity and robustness [5], [4]. Granger Causality is time consuming for evaluating pairwise dependencies when large number of variables are involved. Lasso-Granger [4] is proposed to provide a more scalable and accurate solution, which uses an $L1$-penalty to do variable selection corresponding to dependency discovery. In our work, we use the Lasso-Granger method to discover dependencies among time series from different sensors. The novelty of our work is in extending the notion of dependencies to determine influence of each sensor and using this information in our proposed prediction model for partial data that uses real-time data from only those sensors that have an influence on others.

Big data research in Smart Grid has previously dealt with applications such as optimization and real time forecasting in microgrids using real-time streaming sensor data [6], for clustered time series prediction for customers in utility service areas [33], and for customer selection for demand response using big data analytics [19]. All these works highlight the need to develop approximate algorithms to do scalable analytics on big data witnessed in smart grid. In our work, we highlight that in addition to scalability of models, it is also necessary to have scalability of data collection. In [6], data streams from over 84K sensors per second from buildings in a campus are monitored for predictive analytics, while in [19], data from over 200K smart meters per hour is used for customer selection for demand response. These analyses assume that high resolution data is readily available in real-time as needed. However, at a city-scale, real-time data collection at such large scale from sensors all over a city becomes prohibitive, given the limited capacity of transmission networks. Such scenarios necessitate development of alternative methods - as we do in this paper - that could work with only partial data that is available in real-time.

## III. Preliminaries

In this section, we formulate the problem of prediction with partial data (PPD) addressed in this paper and give formal definitions of key words used in context of the problem. Consider a large set of sensors $\mathcal{S} = \{s_1, ..., s_n\}$ deployed in a big data application. These sensors collect data in real-time[1]. Given network bandwidth constraints, some of these sensors can send data back to a central processing node (CPN) in real-time, while the rest of the sensors send the collected high-resolution readings to the CPN every few hours. Our goal at the CPN is to use this *partial data* to make predictions for *all* sensors for a given prediction horizon $h$.

*Definition 1:* A **time series** output of a sensor $s_i$ is an ordered sequence of readings $\mathcal{T}_i = \{x^i_j\}, j = 1, ..., t$ over past time stamps up to the current time stamp $t$.

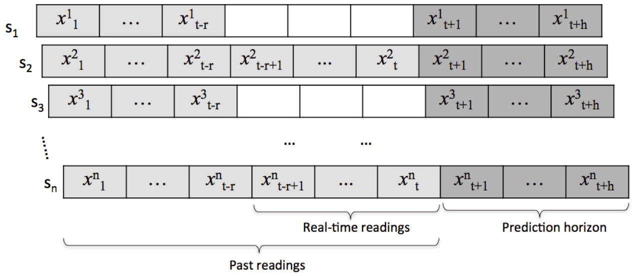

s1
s2
s3
sn
$x^1_1$ … $x^1_{t-r}$ $x^1_{t+1}$ … $x^1_{t+h}$
$x^2_1$ … $x^2_{t-r}$ $x^2_{t-r+1}$ … $x^2_t$ $x^2_{t+1}$ … $x^2_{t+h}$
$x^3_1$ … $x^3_{t-r}$ $x^3_{t+1}$ … $x^3_{t+h}$
… …
$x^n_1$ … $x^n_{t-r}$ $x^n_{t-r+1}$ … $x^n_t$ $x^n_{t+1}$ … $x^n_{t+h}$
Real-time readings
Prediction horizon
Past readings

Fig. 2. Prediction with Partial Data (PPD): Given a set of sensors $S = \{s_1, ..., s_n\}$, some sensors can send readings back to a central processing node (CPN) in real-time, while the rest send collected high-resolution readings back every few hours resulting in partial data at the CPN.

*Definition 2:* Given a set of sensor time series outputs $\{x^i_j\}, j = 1, ..., t, i = 1, ..., n$, **short-term prediction** is to estimate $\{x^i_j\}, j = t+1, ..., t+h, i = 1, ..., n$ for a horizon $h$, which is a few hours ahead.

In this paper, we consider a range of prediction horizons to study how the performance of a prediction model varies with respect to how far the prediction horizon is from the time of prediction. Usually, for short-term predictions, the prediction horizon is within few hours and the prediction intervals are 15-min, 30-min or 1-hour long.

**Problem Definition** We formulate the problem of prediction with partial data (PPD) as follows: Given a set of sensors $\mathcal{S}$ with time series outputs $\{x^i_j\}, j = 1, ..., t, i = 1, ..., n$, make short-term predictions $\{x^i_j\}, j = t+1, ..., t+h, i = 1, ..., n$ for each sensor $s_i \in \mathcal{S}$, when readings $\{x^o_k\}, k = t-r+1, ..., t$ for $o \in \mathcal{O}$ are missing for a subset $\mathcal{O}$ of sensors, $\mathcal{O} \subset \mathcal{S}$.

Figure 2 provides an illustration of the problem. For simplicity, we assume all time-series outputs from sensors to be sampled at the same frequency and be of equal length.

We *hypothesize* that we can learn dependencies in past time series outputs from sensors and use them to make predictions when real-time readings from all sensors is not available. The intuition behind this approach is that many sensors are located spatially close to each other and their readings are likely to be correlated, or they may measure similar entities such as electricity consumption or vehicular traffic which are governed by similar schedules and human activities. If past dependencies can be useful in making future predictions, our next step is to identify the set of sensors that are more helpful in making predictions for other sensors, so that we can collect real-time readings from only these sensors.

*Definition 3:* A **dependency matrix** $\mathcal{M}$ is an $n \times f$ matrix, where each element $\mathcal{M}[i, j]$ represents the dependence of time series $\mathcal{T}_i$ on time series $\mathcal{T}_j$.

The dimensions of a dependency matrix may not be equal. Generally, there may be data available from different types of sensors, while we may want to make predictions for only one type of sensors. In such a case, $n < f$. For example, for making electricity consumption predictions for $n$ smart meters, we may use an additional set of sensors, such as $w$ weather sensors. Then, $f = n + w$. From the dependency matrix, we can now determine which sensors to select for collecting real-time readings.

[1] In context of this paper, data collected at 15-min intervals are considered real-time, even though models proposed in this paper would be equally applicable to data at smaller resolutions.

*Definition 4:* The **influence** $\mathcal{I}^k$ of a time series $\mathcal{T}_k$ is defined as the sum of all values in the column $k$ in the dependency matrix $\mathcal{M}$.

$$\mathcal{I}^k = \sum_{j=1}^{n} \mathcal{M}[j,k] \tag{1}$$

The sensors with higher *influence* values can be selected for collecting real-time readings. This allows transmission of very few readings in real-time compared to the case of transmitting all readings.

*Definition 5:* **Compression Ratio**, $\mathcal{CR}$ is defined as the ratio between the total number of sensor readings that would be required for real-time prediction and the number of readings actually transmitted from selected influential sensors for prediction with partial data.

$$\mathcal{CR} = \frac{\sum_{i=1}^{n} |\mathcal{P}_i|}{\sum_{i=1}^{n} |\mathcal{P}_i| - \sum_{o \in \mathcal{O}} |\mathcal{P}_o|} \tag{2}$$

where $\mathcal{P}_i$ is the sequence of past values from sensor $s_i$ used for prediction and $|\mathcal{P}_i|$ is the length of this sequence; $\mathcal{O}$ is the subset of sensors with missing real-time readings and $n$ is the total number of sensors. For simplicity, we consider same length $l$ of past values for all sensors. Hence, $|\mathcal{P}_i| = l, \forall i$ and above equation can be simplified as

$$\mathcal{CR} = \frac{n}{n - |\mathcal{O}|} \tag{3}$$

Space saving $\mathcal{SS}$ achieved with compression can be correspondingly calculated as: $\mathcal{SS} = 1 - CR^{-1}$.

## IV. Methodology

In this section we describe our proposed method for short-term prediction with partial data. We use a two-stage process, where we first learn dependencies from past data and determine influence for individual sensors, and then use this information for selecting influential sensors for regression tree based prediction.

### A. Influence Discovery

A simple approach to making predictions for a given sensor $s_i \in \mathcal{O}$ in terms of recent real time data from other sensors is to cast it as a regression problem. In an ordinary least squares (OLS) regression, given the data $(\mathbf{x}^i, y_i), i = 1, 2, ..., n$, the response $y_i$ for the $i^{th}$ observation is estimated in terms of $p$ predictor variables, $\mathbf{x}^i = (x_{i1}, ..., x_{ip})$ by minimizing the residual squared error. For our problem of prediction with partial data, the predictor variables for $s_i$ comprise of sequences $\{\mathcal{P}_k\}_{k \neq i}$ of past values from other sensors.

We use a method based on *lasso* for selection of sensors that show stronger *influence* on the given sensor $s_i$ and leave out others. The lasso method is used in regression for shrinking some coefficients and setting others to zero [34]. This is achieved by penalizing the absolute size of the regression coefficients. Another advantage of using the lasso method is that it can improve the prediction accuracy. The OLS method generally gives low bias due to over-fitting but has large variance. By shrinking or setting to zero some of the coefficients, the lasso improves variance and hence may reduce overall prediction errors [34].

Given $n$ sensor outputs in form of time series $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^n$, with readings at timestamps $\mathbf{t} = 1, ...T$, for each series $\mathbf{x}^i$, lasso gives a sparse solution for coefficients $\mathbf{w}$ by minimizing the sum of squared error and a constant times the L1-norm of the coefficients:

$$\mathbf{w} = \arg\min \sum_{t=l+1}^{T} \left\| x_t^i - \sum_{j=1}^{n} \mathbf{w}_{i,j}^T \mathcal{P}_t^j \right\|_2^2 + \lambda \|\mathbf{w}\|_1 \tag{4}$$

where $\mathcal{P}_t^j$ is the sequence of past $l$ readings, i.e., $\mathcal{P}_t^j = [x_{t-l}^j, ..., x_{t-1}^j]$, $\mathbf{w}_{i,j}$ is the $j$-th vector of coefficients $\mathbf{w}_i$ representing the dependency of series $i$ on series $j$, and $\lambda$ is a parameter which determines the sparseness of $\mathbf{w}_i$ and can be determined using cross-validation method. The weight vectors $\mathbf{w}_i$ form the rows of the dependency matrix $\mathcal{M}$. We set diagonals of the dependency matrix to zero, i.e., $\mathcal{M}[i,i] = 0$ in order to remove self-dependencies and simulate the case of making prediction without a sensor's own past real-time data. Given $\mathcal{M}$, *influence* $\mathcal{I}$ of all series can be calculated using equation 1.

### B. Influence Model (IM)

We first split the readings for each smart meter into a set of daily series $\{\mathcal{D}_j^i\}_{i=1,...,n, j=1,...,q}$ at 15-min resolution, where $q$ is the number of days in the data and $n$ is the number of sensors. Splitting into smaller day long windows ensures stationarity for each window. Stationarity means that the dependence on the preceding values does not change with time. Then we learn dependency matrix $\mathcal{M}_j$ for each day as described in Section IV-A.

Predictions for a given day $g$ are based on training data comprising of readings from a previous *similar* day $sim$. We consider two cases of similarity:
1) *previous week*, which is the same day of the week in preceding week, i.e., $sim = g - 7$, if the days are indexed serially. It captures similarity in terms of similar schedules, which for a university campus, and for many other settings involving human activity, follow a weekly pattern.
2) *previous day*, which is the day preceding the given day, i.e., $sim = g - 1$. It captures similarity in terms of weather conditions, as successive days are likely to have little change in weather.

We apply a windowing transformation to the daily series $\{\mathcal{D}^i\}$ in both training and test data to get a set of $\langle$predictor, response$\rangle$ tuples. Given time-series $\mathbf{x}$ with $k$ values, the transformation of length $l$ results in a set of tuples of the form $\langle (x_{t-l+1}, ..., x_t), x_{t+h} \rangle$ such that $l \leq t \leq k - h$.

The prediction model for a sensor $s_i$ is a regression tree [9] that uses predictors from all sensors with non-zero coefficients in the dependency matrix learned from a similar day, i.e., predictors are taken from $\{\mathcal{D}^k\}, \forall k : \mathcal{M}_{sim}[i,k] \neq 0$. Since $\mathcal{M}[i,i] = 0$, sensor $s_i$'s own past values are not used as predictors. Hence, *a key benefit of this model is that we are able to do predictions for a sensor in absence of its own past values by using past values of its influential sensors.*

### C. Local Influence Model (LIM)

In this variant of the influence model, we further reduce the number of sensors used as predictors in the influence model. For each sensor $s_i$, we sort the corresponding row $\mathcal{M}[i,]$ in the dependency matrix and based on this, we consider only readings from the top $\tau_l$ sensors in the influence model. Because we select top influencers *locally* for each sensor, we call this as a local influence model.

### D. Global Influence Model (GIM)

In another variant of the influence model, instead of selecting the top local influencers for each sensor, which when aggregated still require collecting data from a large number of sensors, we propose finding the top set of influencers globally. Using dependency matrices $\mathcal{M}_j$, we calculate daily influence $\mathcal{I}_j^i$ for each sensor $s_i$ as described in equation 1. After sorting the sensors based on their influence values, we consider only readings from the top $\tau_g$ sensors in the influence model. Because we select top influencers *globally* for all sensors, we call this as a global influence model.

## V. Experiments

Here we describe the experiments conducted to evaluate the performance of our models with respect to the baseline model built using real-time data. We also examine how prediction accuracy changes with the number of sensors used for prediction.

### A. Datasets

**Electricity Consumption Data**: This data[2] is collected by smart meters installed in buildings in the USC campus microgrid [32] in Los Angeles. There are over 170 smart meters that collect data at 15-minute intervals. The dataset has about 7 years' electricity consumption values with 170 × 365 × 24 × 4 ≈ 5.96 million readings per year. The average meter reading is 30.5 kWh per 15-min interval, with a standard deviation of 7.65 kWh [2]. In our experiment we use one semester's data from a subset of 115 smart meters that have consistent data output with minimal missing values.

**Weather Data**: This data includes hourly temperature and humidity data taken from NOAA's [26] USC campus/Downtown Los Angeles station. It is linearly interpolated to get 15-min resolution data aligned with the electricity consumption data. We used two more datasets: high temperature data, which retains all temperatures above 70F and sets the rest to zero; and low temperature data, which retains all temperatures below 50F and sets the rest to zero. We use weather data as additional features in our model.

### B. Performance Comparison

In this section we introduce the baseline model and evaluation metrics used in our experiments. We evaluate our models for short term prediction for up to 8 hours ahead prediction horizon[3]. Given the short horizon, the length of previous values is set to 4, equivalent to 1-hour as the data is collected at 15-min granularity. For the baseline as well as the proposed models, we use day-long cross-sections of the data for training and testing, such that training is done on a similar day as the testing day (ref. Section IV-B). In our pilot experiments, we examined two choices of similar day: previous week and previous day, and found previous week to perform better. This can be explained by the similarity in schedules on same weekdays, and hence similar electricity consumption patterns, as opposed to successive days. This observation may be true for many other applications as well that involve sensors collecting data from schedule-related activities. However, sensors collecting other types of information, such as environmental data, may show higher similarity in readings for successive days.

*1) Baseline Model:* As a baseline for other models, we use the Auto-Regressive Tree (ART) Model which uses recent values as features in a regression tree model and has been shown to offer high predictive accuracy on a large range of datasets [25]. For short prediction horizons, recent observations are found to be good indicators of future values. The ART model is a generalization of standard autoregressive (AR) model. It is a piecewise linear model that is learned by recursively dividing the data using a decision tree into smaller areas, where simple linear autoregressive models can be applied. The advantage of ART model lies in its ability to model non-linear relationships in the time-series, which leads to a closer fit to the data than the AR model.

In this paper, we implement a specialized $ART(p, h)$ model that uses recent $p$ values of a variable for making $h$ interval ahead prediction. As our proposed models are also based on similar regression tree concept requiring $h$ interval ahead prediction, the ART model provides a natural baseline to compare our models' performance with. However, it is to be noted that while ART models uses a variable's own recent observations, our models only use other variables' observations to make predictions.

*2) Evaluation Metrics:* We used the Mean Absolute Percentage Error (MAPE) as the evaluation metric to compare different models and strategies. It is defined as:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - \hat{x}_i|}{x_i}$$

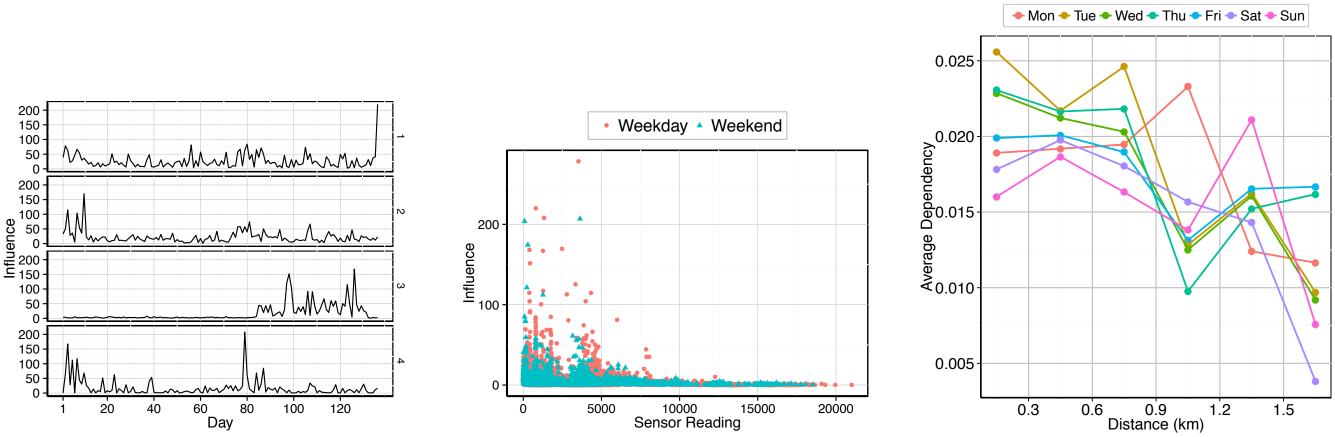
where $x_i$ is the observed value and $\hat{x}_i$ is the predicted value. MAPE was chosen because it is a relative measure and therefore scale-independent [3], which allows comparison across different range of sensor readings.

### C. Influence Variation

In this experiment, we calculated influence for all sensors across all days. Figure 3(a) shows how influence varies for the top 4 influencer sensors on different days. Given how influence for individual sensors varies with time, we decided to recalculate influence for each day in our experiments, rather than use a static value calculated over a large number of days. Next, we examined how size of sensor readings affected its influence. Figure 3(b) shows the distribution of influence for each sensor with the size of sensor readings. It is interesting to note that buildings with smaller consumption values are more numerous and have higher influence. We also observe that influence for

[2] The data is available on request from the USC Facilities Management Services.

[3] For Smart Grid applications such as Demand Response, predictions at 15-min resolution are generally required up to 6 hours ahead [3].



(a)

(b)

(c)

Fig. 3. Variance of Influence/dependency with (a) time, (b) size, and (c) distance: higher values observed for weekdays than for weekends.

weekdays is slightly higher than for weekdays, which could be attributed to more activity of movement of people between buildings on weekdays.

We also investigated how the dependencies (given by the entries in the dependency matrix $\mathcal{M}$) between sensors' time series output varied with the geographical distance between them. In Figure 3(c), we plot average dependency with respect to the distance between pairs of sensors. The results are grouped by averages for each weekday. The distances are binned into 6 bins of 0.3 km each. We observe decreasing dependency as the distance between the sensors increases. This result validates our assumption of greater dependency among sensors located close to each other. In smart grid case, this can be explained by the fact that there is greater movement of people between neighboring buildings compared to those farther apart, and hence more dependency among their electricity consumption. Furthermore, there is more movement on weekdays on campus than on weekends, hence we observe that average dependency is higher for weekdays than for weekends (Figure 3(c)).

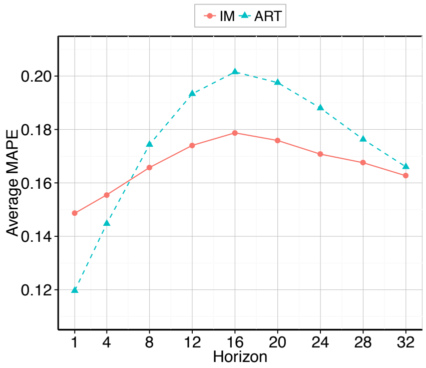
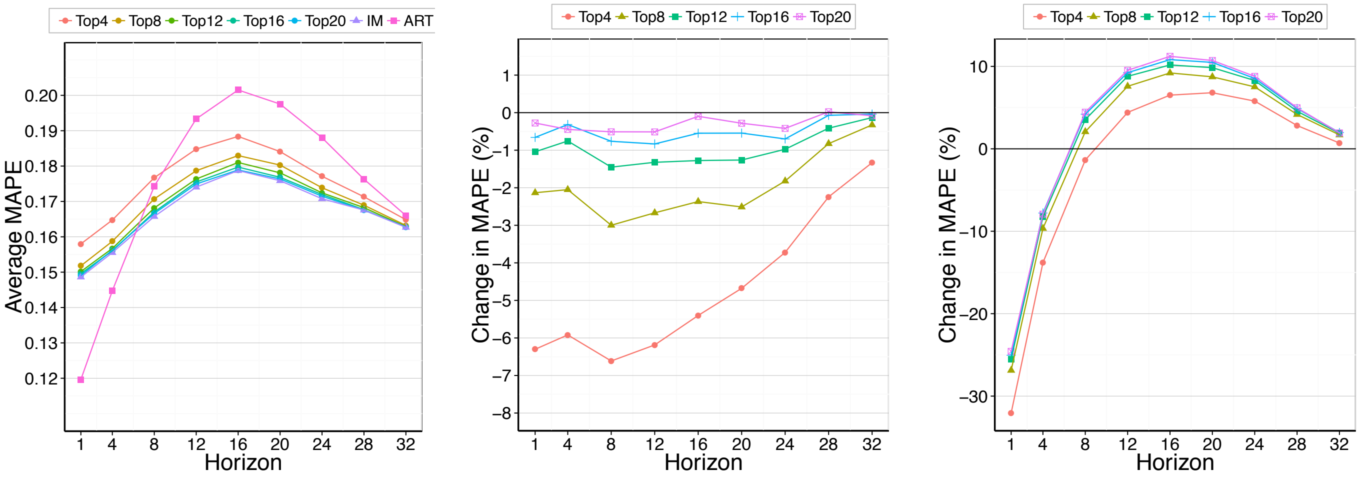


Fig. 4. Performance of influence model (IM) with respect to baseline ART model. For ART, recent values used as predictors at the time of prediction become increasingly ineffective for longer horizons, when IM's use of more recent real-time values of other sensors become more useful predictors.

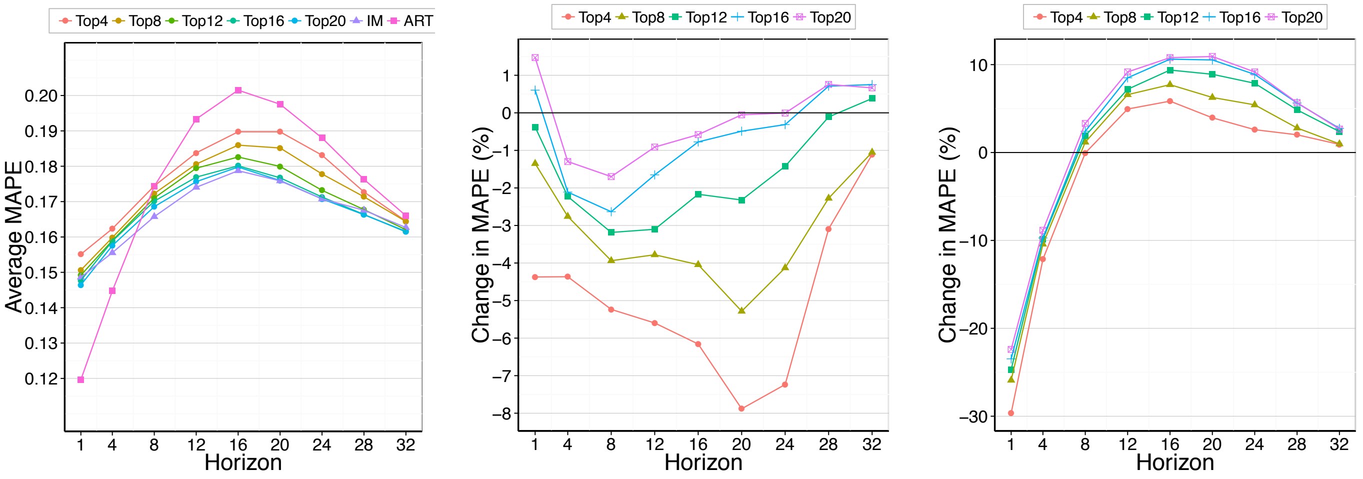
### D. Prediction Performance

In the first set of experiments, we use the *influence model* for making predictions for different prediction horizons up to 8 hours ahead. Figure 4 shows the prediction errors of this model, averaged over all days and for all sensors, with respect to the baseline model. We observe that the baseline ART model performs well up to 6 intervals (1.5 hour). We explain this as an effect of the very-short-term prediction horizon, where electricity consumption is not expected to drastically change from its previous 4 values. Even though influence model does not show clear advantage over the baseline for very-short-term prediction, its accuracy increases with the prediction horizon, where it consistently outperforms the baseline. Also, its advantage becomes clear in light of the fact that it works with only partially available recent values in real-time. Another result to note here is that while increase in IM's error with increasing horizon is more subdued, ART's error increases rapidly with increasing horizon implying that the previous 4 values used as predictors at the time of prediction become increasingly ineffective for predicting values beyond 1.5 hours ahead in time. Here, *more recent real-time values of other sensors actually become more useful predictors than a sensor's own relatively older values*. This is an important result and main advantage of the influence model. Thus, when real-time recent values are not available for a sensor, it uses recent real-time values of other sensors that are identified by learning dependencies among sensors on a similar day in past. In the second set of experiments, we use the *local influence model*, which instead of using all sensors as potential predictors, considers real-time values from only top $\tau$ influential sensors for each sensor. We repeat experiments for $\tau = 4, 8, 12, 16, 20$ for same horizons as used earlier for IM and ART models. In Figure 5(a), we show how the local influence models perform with respect to IM and ART models. As mentioned earlier, we observe ART performing well initially due to very short prediction horizons, but its errors increase rapidly with increasing horizon. *The LIM models show performance comparable to IM, while using real-time values from fewer sensors.* As expected, using increasingly fewer predictors increases the



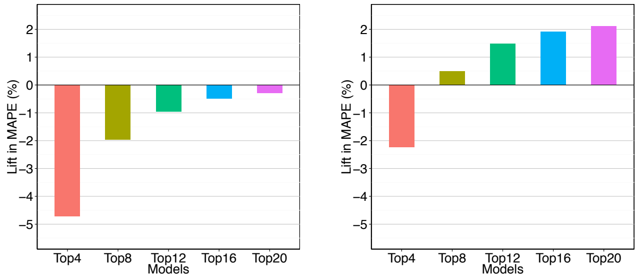
(a) (b) (c)

Fig. 5. Prediction performance of local influence model: (a) Variation in average MAPE for different models; and percentage change in MAPE with respect to (b) IM and (c) ART. (Negative change implies increase in error.)



(a) (b) (c)

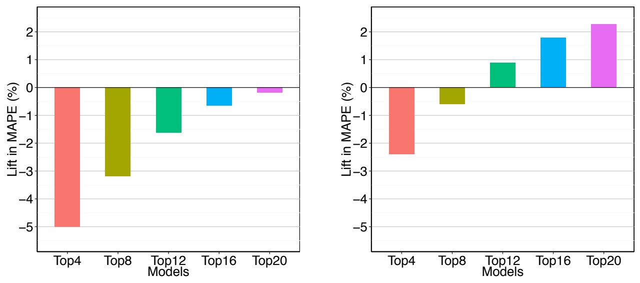
Fig. 7. Prediction performance of global influence model: (a) Variation in average MAPE for different models; and percentage change in MAPE with respect to (b) IM and (c) ART. (Negative change implies increase in error.)



(a) (b)

Fig. 6. Lift in MAPE across all horizons for local influence models with respect to (a) IM and (b) ART. Positive lift is observed w.r.t. ART beyond Top 8. (Positive lift indicates reduction in MAPE.)

prediction error for LIM models, but only slightly. Figure 5(b) shows how LIM models' performance deteriorates compared to IM (in terms of percentage change) and how this gap decreases with increasing horizon. This can be explained as the effect of very few sensors remaining influential over longer horizons. When averaged over all horizons, we observe 4.71% increase in error compared to IM for Top 4 model which comes down to 1.97% increase for Top 8 and less than 1% increase for Top 12, 16, and 20 models (Figure 6(a)). Similarly, we compare performance change in LIM models with respect to ART model in Figure 5(c). We observe that beyond 1-2 hour horizon, all LIM models outperform the ART model, implying that *recent real-time values of a few influential sensors selected locally for each sensor are far more effective as predictors than the sensor's own relatively older values*. When averaged over all horizons, we observe that for Top 4, there is an increase in error by 2.24%, but for Top 8 (and Top 12, 16, 20), the



(a) (b)

Fig. 8. Lift in MAPE across all horizons for global influence models with respect to (a) IM and (b) ART. Positive lift is observed w.r.t. ART beyond Top 12. (Positive lift indicates reduction in MAPE.) In (b) Only $\sim 0.5\%$ increase in prediction error over ART is witnessed while using just top 8 ($\sim 7\%$) of smart meters.

error actually decreases (Figure 6(b)) with respect to ART. Thus, we conclude that for this dataset, we need at least 8 influential sensors for each sensor to improve performance over the baseline model. In the third set of experiments, we use the *global influence model*, which instead of using influential sensors locally for each sensor, uses real-time values from only top $\tau$ influential sensors selected globally for *all* sensors. We repeat experiments for $\tau = 4, 8, 12, 16, 20$ for same horizons as used earlier for ART, IM and LIM models. Figure 7(a) shows the performance of global influence models along with that of IM and ART models. The GIM models outperform the ART model beyond 8 intervals (2-hour horizon), however as the number of predictors is reduced when moving from Top 20 to Top 4 model, we observe that increase in errors is more pronounced for GIM 7(a) than for LIM models 5(a) for same number of predictors. *While LIM used influential sensors selected separately for each sensor, GIM uses the same set of influential sensors for all sensors and still gives comparable performance with only slight deterioration*. Top 20 and Top 16 GIM models even outperform IM (Figure 7(b)) for 1 interval ahead and later for 28 and 32 intervals. This could be due to the large number (20 and 16) of predictors selected in these models overlapping with those of IM models. This is further supported by the average result over all horizons, where both Top 20 and Top 16 models show less than 1% increase in errors compared to IM (Figure 8(a)). We also observe that all GIM models outperform ART beyond 12 intervals, i.e., 3 hour horizon (Figure 7(c)) and require at least 12 influential sensors to improve performance over the baseline across all horizons (Figure 8(b)). Thus, *by sacrificing a fraction of performance accuracy, GIM is able to provide a practical solution using real-time values from only a few global sensors.*

Finally, in Figure 9, we compare average MAPE with respect to compression ratio (Equation 3) to understand the trade-off between prediction accuracy and compression for different prediction horizons. For both LIM and GIM models, we observe that except for very short prediction horizons (1 and 4 intervals), average MAPE either decreases or increases by a very small fraction as compression is increased. This demonstrates the usefulness of influence models which are able to perform well even in absence of recent real-time data from majority of sensors, while using real-time data from only a very small subset of sensors.

## VI. Conclusion and Future Work

In this paper, we address the issue of data *veracity* in big data applications that arises when real time data from all sensors is not available at central nodes due to network latency or faults, or when limited by consumers for security and privacy reasons. Standard models for short term predictions are either unable to predict or perform poorly when trying to predict with *partial data*. We introduce novel *influence* based models to make predictions in absence of real-time data from majority of sensors using real-time data from only a few influential sensors. Our experimental results on real world data derived from the smart grid domain indicate that the performance of our models is comparable to those of baseline autoregressive tree model. In certain cases, they are also able to add extra predictive accuracy. Thus, these models provide a practical alternative to canonical methods for dealing with missing real-time readings in sensor streams, which is generalizable to big data applications in several domains, such as in healthcare applications, environment monitoring, and in smart urban infrastructure. These models provide a simple and interpretable solution that is also easy to understand and apply for domain experts.

Future extensions of this work require further investigation into scenarios in which the different approaches discussed in the paper are most effective, and the range of big data problems for which they can be useful. Another direction of research is towards a two stage process for influence discovery, guided by some heuristics, to make the process more scalable. Further research is also required to improve the performance of these methods, possibly by performing a combination of local and global selection of influential sensors.

## Acknowledgment

This material is based upon work supported by the United States Department of Energy under Award Number DEOE0000192, and the Los Angeles Department of Water and Power (LA DWP). The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, the LA DWP, nor any of their employees. The authors would like to thank Marc Frincu, Anand Panangadan and Om Patri for their inputs and feedback in preparing this paper.

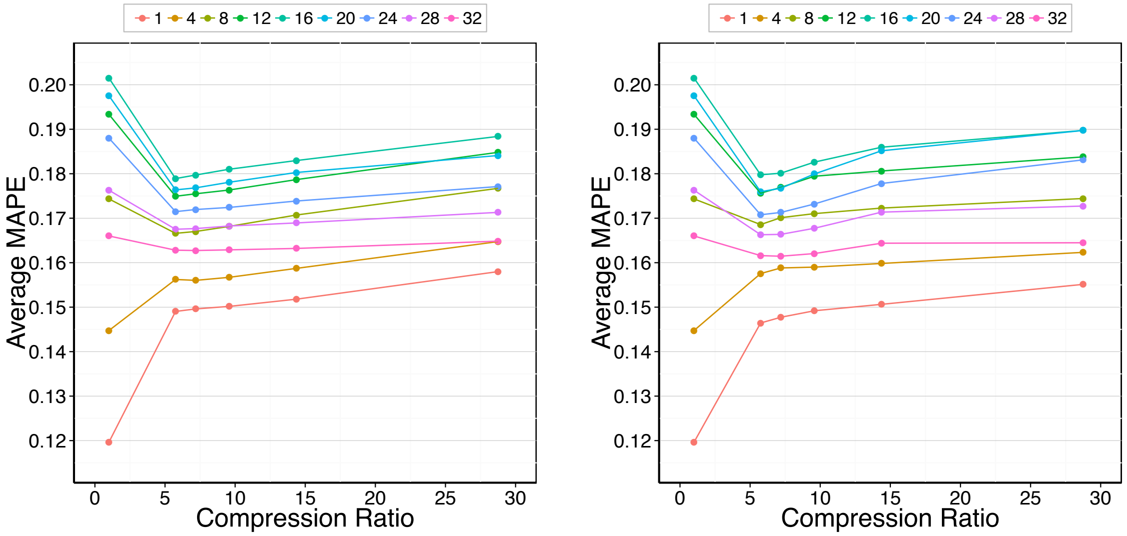
## References

[1] N. Addy, S. Kiliccote, J. Mathieu, and D. S. Callaway. Understanding the effect of baseline modeling implementation choices on analysis of demand response performance. In *ASME 2012 International Mechanical Engineering Congress and Exposition*, 2013.

[2] S. Aman, M. Frincu, C. Chelmis, M. U. Noor, Y. Simmhan, and V. Prasanna. Empirical Comparison of Prediction Methods for Electricity Consumption Forecasting. *CS Department Technical Report 14-942, University of Southern California*, 2014.

[3] S. Aman, Y. Simmhan, and V. Prasanna. Holistic measures for evaluating prediction models in smart grids. *IEEE Transactions in Knowledge and Data Engineering (To appear)*, 2014.

[4] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *International conference on Knowledge discovery and data mining (KDD'07)*. ACM, 2007.



(a) (b)

Fig. 9. Trade-off between prediction accuracy and compression for different horizons for (a) local influence model and (b) global influence model

[5] M. T. Bahadori and Y. Liu. Granger causality analysis in irregular time series. In *SIAM International Conference on Data Mining (SDM 2012)*. SIAM, 2012.

[6] N. Balac, T. Sipes, N. Wolter, K. Nunes, R. S. Sinkovits, and H. Karimabadi. Large scale predictive analytics for real-time energy management. In *IEEE International Conference on Big Data*, 2013.

[7] F. Bouhafs, M. Mackay, and M. Merabti. Links to the future: communication requirements and challenges in the smart grid. *IEEE Power and Energy Magazine*, 10(1), 2012.

[8] G. E. P. Box and G. M. Jenkins. *Time series analysis, forecasting and control*. Holden-Day, 1970.

[9] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[10] K. Bullis. Smart wind and solar power. *MIT Technology Review*, 2014.

[11] C. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91, 2003.

[12] A. Ciancio and A. Ortega. A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, 2005.

[13] J. C. Cuevas-Tello, P. Tiňo, S. Raychaudhury, X. Yao, and M. Harva. Uncovering delayed patterns in noisy and irregularly sampled time series: an astronomy application. *Pattern Recognition*, 43(3), 2010.

[14] Y. Demchenko, P. Grosso, C. de Laat, and P. Membrey. Addressing big data issues in scientific data infrastructure. In *International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, 2013.

[15] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *International conference on Very large data bases (VLDB 2004)*, 2004.

[16] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[17] B. Karimi, V. Namboodiri, and M. Jadliwala. On the scalable collection of metering data in smart grids through message concatenation. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2013.

[18] D. M. Kreindler and C. J. Lumsden. The effects of the irregular sample and missing data in time series analysis. *Nonlinear dynamics, psychology, and life sciences*, 10(2), 2006.

[19] J. Kwac and R. Rajagopal. Demand response targeting using big data analytics. In *IEEE International Conference on Big Data*, 2013.

[20] D. Laney. 3D Data Management: Controlling data volume, velocity and variety. *Gartner*, 2001.

[21] A. C. Lozano, H. Li, A. Niculescu-Mizil, Y. Liu, C. Perlich, J. Hosking, and N. Abe. Spatial-temporal causal modeling for climate change attribution. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. ACM, 2009.

[22] A. Marascu, P. Pompey, E. Bouillet, O. Verscheure, M. Wurst, M. Grund, and P. Cudre-Mauroux. Mistral: An architecture for low-latency analytics on massive time series. In *IEEE International Conference on Big Data*, 2013.

[23] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *IEEE Security and Privacy*, 7, 2009.

[24] E. McKenna, I. Richardson, and M. Thomson. Smart meter data: Balancing consumer privacy concerns with legitimate applications. *Energy Policy*, 41(C), 2012.

[25] C. Meek, D. M. Chickering, and D. Heckerman. Autoregressive tree models for time-series analysis. In *2nd International SIAM Conference on Data Mining (SDM)*. SIAM, 2002.

[26] NOAA. Quality Controlled Local Climatological Data Improvements/Differences/Updates. 2013.

[27] B. Pan, U. Demiryurek, and C. Shahabi. Utilizing real-world transportation data for accurate traffic prediction. In *IEEE International Conference on Data Mining (ICDM'12)*, 2012.

[28] M. A. Razzaque, C. Bleakley, and S. Dobson. Compression in wireless sensor networks: A survey and comparative evaluation. *ACM Transactions on Sensor Networks*, 10(1), 2013.

[29] B. Saha and D. Srivastava. Data quality: The other face of big data. In *IEEE International Conference on Data Engineering (ICDE)*, 2014.

[30] P. Sanders, S. Schlag, and I. Muller. Communication efficient algorithms for fundamental big data problems. In *IEEE International Conference on Big Data*, 2013.

[31] A. Shojaie and G. Michailidis. Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18), 2010.

[32] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna. Cloud-based software platform for data-driven smart grid management. *IEEE Computing in Science and Engineering*, 2013.

[33] Y. Simmhan and M. U. Noor. Scalable prediction of energy consumption using incremental time series clustering. In *IEEE International Conference on Big Data, Workshop on Big Data and Smarter Cities*, 2013.

[34] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1), 1996.