A Survey About Prediction-Based Data Reduction in Wireless Sensor Networks

GABRIEL MARTINS DIAS, BORIS BELLALTA, and SIMON OECHSNER, Pompeu Fabra University

One of the main characteristics of Wireless Sensor Networks (WSNs) is the constrained energy resources of their wireless sensor nodes. Although this issue has been addressed in several works and received much attention over the years, the most recent advances pointed out that the energy harvesting and wireless charging techniques may offer means to overcome such a limitation. Consequently, an issue that had been put in second place now emerges: the low availability of spectrum resources. Because of it, the incorporation of the WSNs into the Internet of Things and the exponential growth of the latter may be hindered if no control over the data generation is taken. Alternatively, part of the sensed data can be predicted without triggering transmissions that could congest the wireless medium. In this work, we analyze and categorize existing prediction-based data reduction mechanisms that have been designed for WSNs. Our main contribution is a systematic procedure for selecting a scheme to make predictions in WSNs, based on WSNs' constraints, characteristics of prediction methods, and monitored data. Finally, we conclude the article with a discussion about future challenges and open research directions in the use of prediction methods to support the WSNs' growth.

CCS Concepts: • **Networks** → **Sensor networks**; • **Computer systems organization** → **Sensor networks**; • **Information systems** → Sensor networks

Additional Key Words and Phrases: Predictions, wireless sensor networks, data science, data reduction, machine learning

ACM Reference Format:
Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner. 2016. A survey about prediction-based data reduction in wireless sensor networks. ACM Comput. Surv. 49, 3, Article 58 (November 2016), 35 pages.
DOI: http://dx.doi.org/10.1145/2996356

1. INTRODUCTION

Wireless sensor nodes (sensor nodes, for brevity) are small computer devices with low production costs, equipped with a radio antenna and sensors that are capable of sensing one or more environmental parameters [Akyildiz et al. 2002]. Thanks to their portable size, sensor nodes are often densely deployed in areas that may not be humanly accessible. Hence, one of the biggest challenges of working with battery-equipped sensor nodes has been their limited energy availability, which is compounded by the fact that (1) radio transmissions are the operations that consume the most energy and (2) Wireless Sensor Networks (WSNs) are mainly data-oriented networks; that is, their most valuable asset is the data that sensor nodes can produce.

This work has been partially supported by the Spanish government through the project TEC2012-32354 (Plan Nacional I+D), by the Catalan government through the project SGR-2014-1173, and by the European Union through the project FP7-SME-2013-605073-ENTOMATIC.

Authors' addresses: G. M. Dias; email: gabriel.martins@upf.edu; B. Bellalta; email: boris.bellalta@upf.edu; S. Oechsner; email: simon.oechsner@upf.edu.

DOI: http://dx.doi.org/10.1145/2996356

Nonetheless, a recent survey presented in Rault et al. [2014] listed several efforts to manage the energy consumption of WSNs at different levels. In this survey, among several applications and routing and energy-saving techniques, there are promising advances in the energy supply methods for sensor nodes, including energy harvesting [Sudevalayam and Kulkarni 2011] and wireless power transfer [Xie et al. 2013]. According to Rault et al. [2014], wireless power charging facilitates the design of scalable methods to refill network elements' batteries, allowing the sensor nodes' energy constraint to be overcome. Meanwhile, the medium access is one of the key challenges in the next generations of wireless networks due to the increasing number of wireless devices and different traffic profiles [Bellalta et al. 2016]. Therefore, we foresee the urgency of reducing the number of transmissions to support the growth in the number of wireless devices, and the use of predictions is a promising alternative, also because they have potential as energy-saving mechanisms. For the presented reasons, we survey existing approaches that use predictions to reduce the number of transmissions in WSNs.

Indeed, many prediction-based data reduction techniques have been designed for minimizing their radio energy consumption, without concerning the medium access limitations. For example, the survey presented in Anastasi et al. [2009] described several methods and architectures used to reduce the energy consumption and extend the WSNs' lifetime. There, works are labeled according to their main characteristics, including a *data-driven* category that encompasses a specific set of works focused on *data prediction*. As they focus on WSNs' energy consumption, some categories overlap in terms of how predictions are adopted and computed in WSNs. Furthermore, as the authors highlighted, by the time of the publication, forecasting methods had not been fully explored in WSNs and only a very limited number of algorithms had been adopted, because high-complexity methods were thought to be unsuitable for sensor nodes. Later, such an assumption started to be challenged, and real deployments incorporated advanced forecasting methods [Aderohunmu et al. 2013b] and other artificial intelligence tools [Askari Moghadam and Keshmirpour 2011]. More recently, in Rault et al. [2014], several techniques to reduce the energy consumption in WSNs were compared according to the requirements of WSN applications. Once again, works have been analyzed from the energy consumption perspective and, besides including several approaches that do not use predictions, the authors focus on neither how the predictions are adopted in WSNs nor which prediction methods are used to reduce the data.

Data reduction methods encompass different techniques that may lower the number of transmissions, but not all of them involve predictions. For example, the mechanism presented in Deligiannakis et al. [2011] prioritizes routes through sensor nodes that are collecting data at a certain time and paths that can aggregate more information thanks to the data similarity. Hence, WSNs' routing topology is optimized in favor of reducing their energy consumption and number of transmissions. Alternatively, Intanagonwiwat et al. [2001] shows that a simple aggregation scheme that joins the data from packets and suppresses their headers' information can efficiently reduce the number of transmissions in the WSNs. The survey presented in Luo et al. [2007] contains other examples of intelligent routing schemes, Fasolo and Rossi [2007] list other data aggregation methods, and Srisooksai et al. [2012] provide a survey of mechanisms for data compression. All of these application types can reduce the number of transmissions, but prediction-based data reduction methods are not restricted to only one of them, as we will exemplify in Section 3.

Over the years, with the evolution of hardware, some works started shifting the paradigm of avoiding complex algorithms in WSNs. The works surveyed in Mahmood et al. [2013] take into account the WSNs' constraints to adopting data mining techniques aiming to find patterns in the sensor data to improve its collection and delivery. For instance, if nodes that measure similar information integrate the same cluster, redundant data can be efficiently suppressed, increasing the overall data delivery and improving the WSN's energy efficiency [Guo et al. 2009]. Even though some techniques adopted in data mining also include predictions, they are mainly applied to extract relevant information hidden in the sensed data, and many works included in the survey do not address the reduction in the number of transmissions. Similarly, Alsheikh et al. [2014] discussed the adoption of machine-learning techniques at different layers, such as routing, medium access control, and event detection. In our work, a machine-learning technique, called Artificial Neural Networks (ANNs), will also be presented. However, we also consider many statistical and probabilistic methods, such as the traditional time series methods (autoregressive and moving average methods), that are not considered machine learning. While machine-learning techniques rely on their ability to learn and evolve their predictions in response to environment changes, traditional time series methods rely on the statistics of the studied data to make predictions.

For the reasons explained at the beginning, we limit the scope of this survey to mechanisms that use predictions as a means to reduce the number of transmissions in a WSN. Moreover, in our work, we focus not only on presenting the current solutions for WSNs but also on introducing existing prediction techniques, featuring methods that are currently being used in data reduction solutions in WSNs. To the best of our knowledge, this is the first time that this approach has been taken.

The rest of this article is organized as follows: In Section 2, we explain the terms and jargon that will be used to characterize WSNs and predictions in the rest of the work. The following sections list the works that use predictions to reduce the number of transmissions, according to the structure shown in Figure 1: In Section 3, we introduce and explain the Single Prediction Schemes (SPSs), featuring where the predictions are computed; and in Section 4, we detail the Dual Prediction Schemes (DPSs). Section 5 shows the current state of the art from the perspective of the data predictions, that is, which are the methods used to make predictions in WSNs, as well as the advantages and disadvantages of each one. Later, we provide a discussion of the main questions and challenges observed in the surveyed works in Section 6, before listing, in Section 7, the issues that are still open in this area and the guidelines for new works that intend to solve such problems and improve the state of the art. Finally, we draw the conclusions in Section 8.

2. FUNDAMENTALS OF WSNS AND DATA PREDICTION

In order to make it clearer for readers, we adopt a standard set of terms and describe all the surveyed works using the same nomenclature. In this section, we present the default representation considered for WSNs and explain the terms used to describe predictions, which may be crucial for understanding the rest of this work.

2.1. WSN Organization

A typical WSN is composed of dozens (occasionally hundreds) of ordinary sensor nodes connected to a central workstation that is responsible for providing the communication between the WSN owner and the sensor nodes. This link is bidirectional: it can be used to inform the reported data to the WSN owner and to (re-)configure the sensor nodes' operation. From now on, we will refer to this central workstation as *Gateway* (GW).

In some cases, the excessive number of sensor nodes demands an internal WSN reorganization to avoid packet losses and reduce the number of packet collisions. Such an internal organization, which facilitates the communication between sensor nodes and saves their batteries, is made by "clustering" sensor nodes according to their location or according to the correlation between their measurements (see Abbasi and Younis [2007] for further details). As shown in Figure 2(a), a WSN may be organized in one or several clusters.

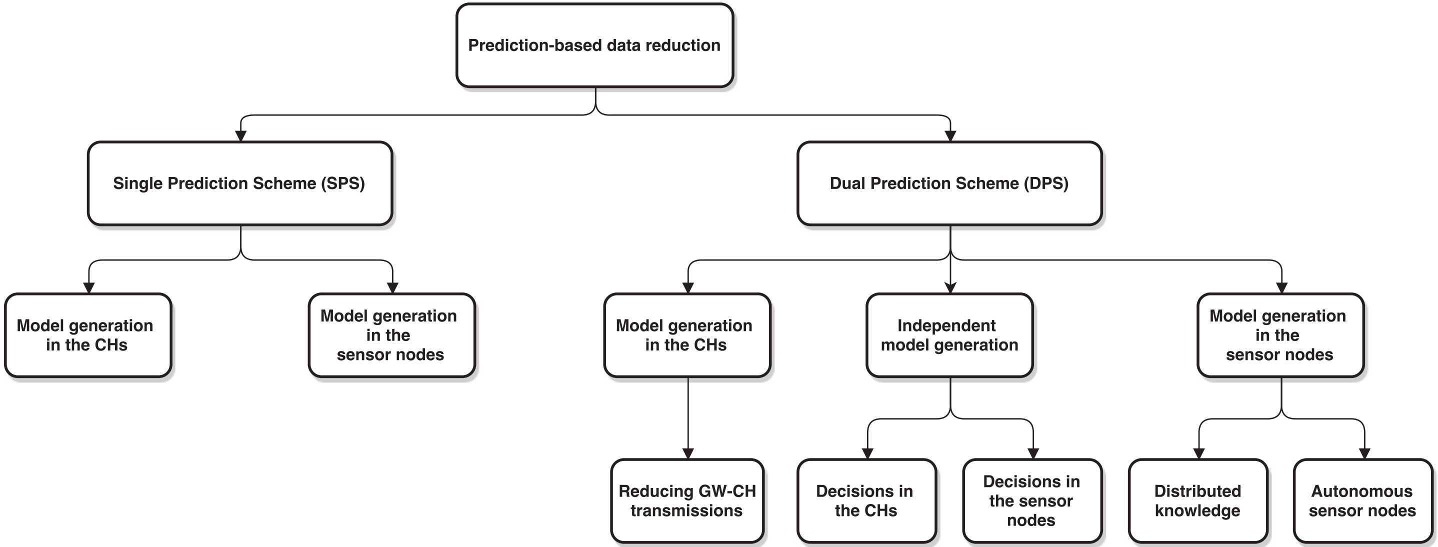
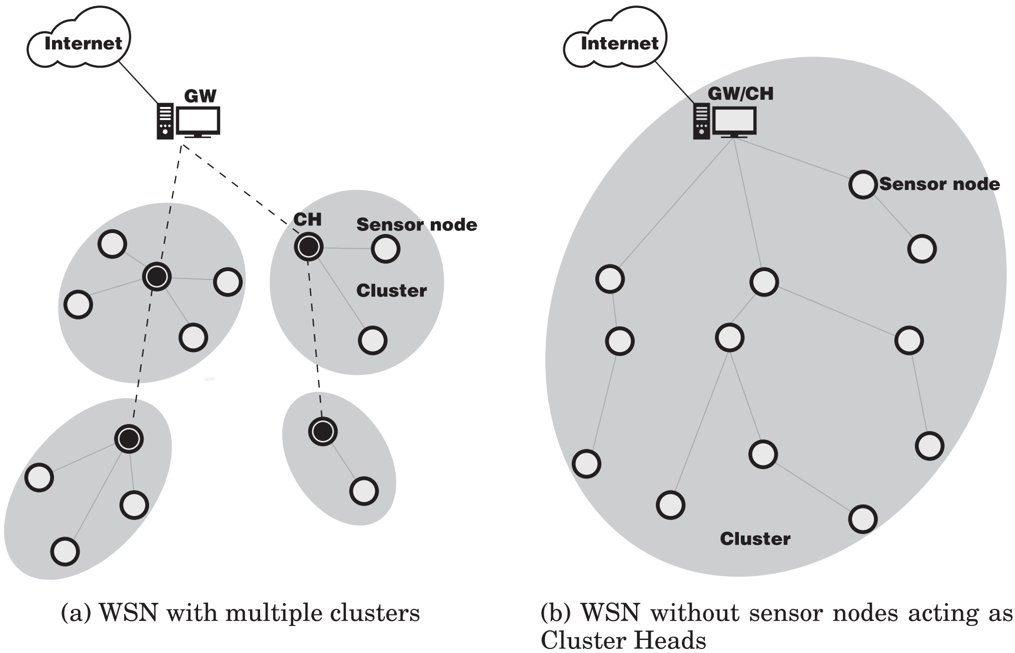

Prediction-based data reduction
Single Prediction Scheme (SPS)
Dual Prediction Scheme (DPS)
Model generation in the CHs
Model generation in the sensor nodes
Model generation in the CHs
Independent model generation
Model generation in the sensor nodes
Reducing GW-CH transmissions
Decisions in the CHs
Decisions in the sensor nodes
Distributed knowledge
Autonomous sensor nodes

Fig. 1. Taxonomy of the architectures that use predictions for data reduction.



(a) WSN with multiple clusters

(b) WSN without sensor nodes acting as Cluster Heads

Fig. 2. Typical WSN scenarios and their roles.

In a cluster, the communication between the sensor nodes and the GW is a responsibility of the Cluster Head (CH). For example, CHs must inform sensor nodes about decisions taken by the WSN owner and transmit the sensed data to the GW. As the sensor nodes in a cluster are usually near each other, it is easier for CHs to keep the control of transmissions and reduce the sensor nodes' energy wasting. Most of the works considered in this survey do not focus on ways to cluster sensor nodes, but they usually rely on existing methods to assume a cluster-based organization.

Alternatively to clustered WSNs, a flat organization may be adopted, as shown in Figure 2(b). In such cases, GWs can communicate directly with sensor nodes and therefore assume the responsibility for establishing the communication between sensor nodes and WSN owners. As CHs and GWs have the similar responsibility of gathering data from sensor nodes and pushing it forward, from now on, we will refer to GWs as CHs in order to facilitate understanding for the reader. In specific cases where the communication between GWs and CHs is discussed, it will be clearly stated.

2.2. Data Prediction

The term *prediction* can refer either to the process of inferring missing values in a dataset based on statistics or empirical probability or to the estimation of future values based on the historical data. A *prediction method* (P) is a function that produces predictions based on two input values: a set of observed values (X) and a set of parameters (θ). A *prediction model* (p) is an instance of a prediction method P, such that $p_\theta(X) = P(X, \theta)$; that is, every prediction model is deterministic, and its output depends only on the set of observed values. The values of θ can be chosen based on the evaluation provided by a utility function that can measure predictions' accuracy, models' complexity, and information loss. In conclusion, it is possible to create different prediction models that use the same algorithm (i.e., the same prediction method).

A prediction method may require some information about the data that is going to be predicted, for example, the assumption that the values will be normally distributed. In some cases, this knowledge is already owned by the user before the deployment of the WSN and can be applied to statistical methods, such as linear regressions (see Timm [2002] for further details). The positive aspect of statistical methods is that it is possible to estimate the yield of the system beforehand. For example, based on the assumption about the data normality, the probability of making accurate predictions can be used to calculate whether the gains that the system can achieve will be worth the investment to be done.

On the other hand, machine-learning techniques require fewer assumptions about the data in exchange for a period to adjust their parameters and adapt to the data that is being monitored, as described in Haykin [2009]. As a drawback, it is not possible to estimate how the system will perform in the real world before it is actually running. Usually, these techniques are tested through real data and adapt to occasional changes, but no guarantees about the predictions' accuracy can be given.

3. SINGLE PREDICTION SCHEMES

In the Single Prediction Schemes (SPSs), predictions are made in a single point of the network, which can be either close to the origin of the data (in sensor nodes) or close to the data collection point (in CHs). For instance, CHs can predict the data measured by sensor nodes and autonomously decide when to pull more measurements based on the reliability of the predictions. Alternatively, sensor nodes can predict changes in their surroundings to avoid unnecessary measurements and—consequently—their transmissions. The latter option is especially beneficial if a sensor node spends more energy to sample the environment than to compute a set of machine instructions that will predict the future measurements.

The main advantage of SPSs is that each device can decide by itself whether to adopt predictions or not, and there is no overhead to communicate about their decisions or synchronize with their neighbors. As a drawback, there is an eventual reduction in the quality of the information provided by CHs, given that WSNs resign part of the data generated by their sensors. In this section, we categorize, according to the place where the predictions are made, existing works that adopt SPSs to reduce the number of transmissions.

3.1. Model Generation in the CHs

As CHs usually have higher computational power and energy availability, they can locally generate prediction models and make important decisions about the WSNs' operation without compromising the quality of the information provided by the measurements. On the other side, a conservative strategy is adopted in sensor nodes, which become merely responsible for their primary tasks, that is, measuring environmental parameters and transmitting the raw data collected by their sensors.

Especially in environmental monitoring WSNs, measurements made by closely positioned sensor nodes have a spatiotemporal correlation, which can be used to generate probabilistic models, approximate the data to well-known distributions, and associate confidence levels to predictions. Hence, the number of transmissions can be reduced if CHs predict measurements and locally check whether the user-imposed quality constraints are matched or not. Because of the autonomy given to the CHs, this scheme has been used in several application types, such as adaptive sampling, clustering, and data compression.

Adaptive Sampling. Generating prediction models in CHs can be an efficient method to answer queries without fetching the data directly from the sensor nodes, as shown in Cheng et al. [2003]. User queries contain, besides the data that should be returned, the error tolerated by the user. Therefore, CHs can answer that the actual current measurements are inside a range of values if their confidence is high enough to satisfy a user-tolerated error. To achieve that, CHs must be able to compute prediction models based on the statistics of the historical data—considering the uncertainty about the current values—and autonomously decide whether to pull more measurements or not. As an alternative, CHs can use inferential statistics to decide which sensor nodes have to be sampled, based on their odds of providing valuable information to the user.

In Deshpandem et al. [2004], the mechanism called BBQ adopted linear regressions to exploit the correlation between different types of data that the sensor nodes may be able to measure, for example, their own voltage and the local temperature. Simulations using real data show that the mechanism can reduce the number of transmissions, save energy, and keep a high confidence level (95%) about the information retrieved. Moreover, its was possible to keep a low number of mistakes in a scenario with little human intervention, that is, where the environment is influenced by fewer external factors.

More recently, Principal Component Analysis (PCA) was used to analyze the historical data and select only the sensor nodes that measured most of the variance observed in the environment [Malik et al. 2011]. The latter technique reduced the workload of the sensor nodes and prolonged twice the WSN lifetime, according to the results obtained from experiments in real testbeds.

Topology Control. The works done in Emekci et al. [2004] and Yann-Ael and Gianluca [2005] exploit the spatiotemporal correlation between the sensor nodes' measurements to build sets of nodes that can provide "trustful" measurements and should be regularly sampled. In practice, only a subset of sensor nodes is activated during a time interval and all the others have their radios and sensors turned off to reduce the number of transmissions, save energy, and extend the WSN lifetime. In order to fairly extend the WSN lifetime, every sensor node must be queried at least once during a cycle, and the number of times that they are activated in a cycle depends on the remaining energy on each one's battery. On the other hand, every subset of sensor nodes provides the values used to predict the measurements of the whole WSN. The predicted values, on average, should differ by less than a user-defined threshold from values obtained when using measurements from all the sensor nodes. The Binocular framework (presented in Emekci et al. [2004]) defines before the WSN deployment the subsets of sensor nodes that must be active at a time. During the so-called *data processing* phase, CHs receive measurements from sensor nodes and calculate linear transformations to make the predictions based on those sensor nodes that will remain active. On the other hand, in Yann-Ael and Gianluca [2005], measurements are assumed to follow normal distributions. Simulations using real data show that their approach can be used to extend the WSNs' lifetime when the requirements about the accuracy are not very strict, namely, when the temperature can be wrong by $\pm 0.5^{o}$C with a confidence level of 0.95. Furthermore, WSNs must be dense enough so that some sensor nodes can be switched off and their measurements inferred using their neighbors' measurements.

Clustering. The algorithm presented in Tulone and Madden [2006] introduced a method to build clusters and aggregate sensed data based on their similarity. In short, nodes are accepted as part of a cluster if their measurements are similar to their neighbors' measurements, which reduces their divergences and the deviation from the average values, and makes the data easier to compress, for example. Simulations using data collected by real sensor nodes showed that it is possible to reduce the number of transmissions done in the network without injecting significant errors to the reported data. The main drawback of this mechanism is that, as the sensor nodes' roles in the clusters rely on data analysis, it is not possible to assign the role of CH based on the availability of resources, such as higher energy availability or higher computational power to perform advanced instructions. This limitation impacts the solution presented in Yin et al. [2015], which uses PCA to reduce the number of dimensions of the data and transmit less data from CHs to GWs. That is, CHs may not have the computational power required to run PCA, because this method has high complexity and relies on some advanced instructions that cannot be performed in the simplest wireless sensor nodes, such as the multiplication of large matrices.

3.2. Model Generation in the Sensor Nodes

If the prediction models are generated in the sensor nodes and not shared with the CHs, their computing tasks may go far beyond simple data reporting. For example, without the CHs' intervention, they can decide if a measurement has to be made (or transmitted) based on the quality of the information that it can provide.

Given that the sensor nodes' computing power can be constrained, decisions about predictions may be supported by their neighbors' data; that is, they can be distributed. For instance, instead of transmitting every measurement to the CH, a sensor node may locally decide not to transmit after observing that its neighbors' measurements are sufficient to accurately monitor its region. WSNs used for event detection and object tracking may predict in the sensor nodes to avoid burst transmissions that could provoke packet losses and delay the delivery of important messages to the CHs.

Furthermore, WSNs for object tracking are usually composed of more powerful and reliable sensors, such as cameras, microphones, and radio-frequency identification [Bhatti and Xu 2009], in which the sampling process generally consumes more energy than the traditional temperature and relative humidity monitoring devices [Anastasi et al. 2009]. Hence, predictions can be used to adjust their sampling rate and avoid unnecessary measurements, as proposed in the Prediction-based Energy Saving (PES) scheme [Xu et al. 2004b]. The mechanism defined by the PES has three main components: (i) a simple prediction model that can be computed by the sensor nodes and avoid unnecessary computation, (ii) a wake-up mechanism that defines which nodes should be turned on after making a prediction about where the object is going to be in the next time interval, and (iii) a recovery mechanism, in order to turn all sensor nodes on whenever an object that is expected to be in the range of WSN cannot be found by the active ones. The PES scheme aims to minimize the miss rate while tracking objects and to maximize the energy savings in the WSN, which is achieved by reducing the sensor nodes' computing time and their number of transmissions. Simulation results showed that the success of this scheme depends mostly on the number of objects that will be tracked at the same time. Once this information is known, the dynamics of their movements play a major role in the system's efficiency. In Samarah et al. [2011], the PES scheme was simulated using a multidimensional regression analysis to predict the movements of the tracked objects. The simulation results showed that it was possible to keep a low energy consumption level while maintaining the missing rate less than 20%.

4. DUAL PREDICTION SCHEMES

In Dual Prediction Schemes (DPSs), the predictions are simultaneously made in CHs and sensor nodes. The general idea behind such mechanisms is that sensor nodes are able to produce the same "a priori" knowledge as CHs are, but sensor nodes can locally check the predictions' accuracy and avoid unnecessary transmissions. As shown in Figure 3, the same prediction model is shared between the sensor node and its respective CH. Then, every time the sensor node measures a value that falls outside

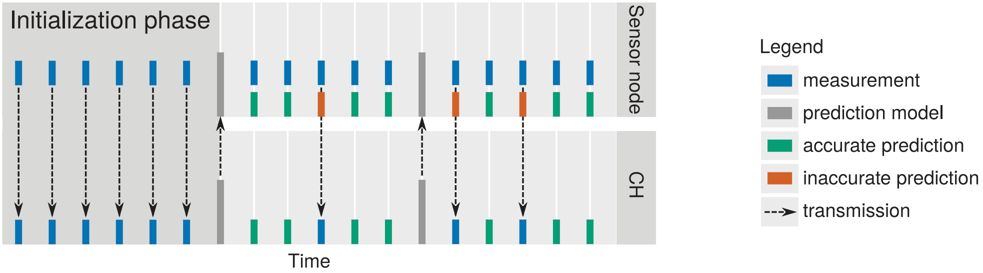


Fig. 3. In a DPS, a measurement is transmitted only if its forecast is inaccurate. The CHs may be responsible for transmitting new prediction models every time interval after the initialization phase.

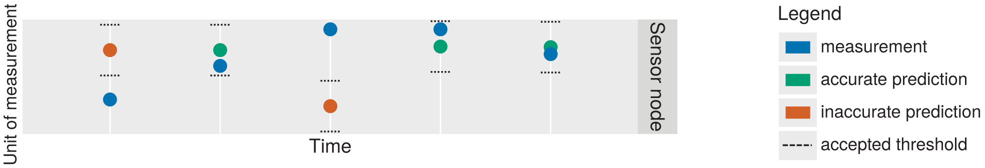


Fig. 4. Measurements that fall inside the accepted threshold do not trigger any action.

an acceptance threshold defined for the predictions (as represented by the first and the third measurements in Figure 4), it must transmit the real value to its CH, which substitutes the local predictions by the correct value. Hence, sensor nodes can consume fewer energy resources and avoid unnecessary transmissions, because measurements will be transmitted to the CHs only when the predictions are not sufficiently accurate.

Prediction models can be generated in CHs and further shared with sensor nodes, or vice versa (for instance, the work presented in Lazaridis and Mehrotra [2003] allows both approaches). Once in a while, a new prediction model may be generated if the current one is not predicting as accurately as expected. As DPSs aim to reduce the number of transmissions without compromising the quality of information generated by the WSN, they target the tradeoff between the number of transmissions and the quality of measurements provided by the system. In this case, DPSs' efficiency depends not only on predictions' accuracy but also on the number of transmissions required to distribute new prediction models and on the channel's reliability. For instance, a high bit error rate may result in the absence of updates arriving at the CHs, which is usually treated as a signal of predictions' high accuracy. Hence, to avoid that, packets will have to be often retransmitted, congesting the medium, consuming extra energy, and diminishing the theoretical gains. Therefore, to decide for a new prediction model, it is necessary to observe what is the most proper prediction method, given the current environmental conditions, and if making predictions (instead of transmitting all measurements) will reduce the number of transmissions in the WSN. These observations may be made either in CHs or in sensor nodes, independently of their responsibility to generate new prediction models, in order to keep the high accuracy and the scheme's yield.

Alternatively, a sensor node and its CH may generate the same prediction model, independently and at the same time, without the necessity of generating extra transmissions in the WSN. This requires a previous knowledge about the environment and the data that is going to be measured, in order to program and configure sensor nodes to decide for the same methods as CHs will adopt in runtime. Furthermore, in this case, the predictions' accuracy may be restricted by the sensor nodes' computing capacity, because sensor nodes may not be able to adopt prediction methods that require more memory, storage, or processing power.

4.1. Model Generation in CHs

Generating the prediction models in CHs exploits the asymmetric computational power availability in WSNs: CHs usually have cheaper energy sources and more resources (such as memory and processing power) than ordinary sensor nodes that are mainly used for measuring and reporting environmental data. As presented in Goel and Imielinski [2001] and Liu et al. [2005], in the beginning, sensor nodes transmit the current measurements to their CHs. Based on received values, CHs could locally generate new prediction models for each sensor node. Thus, CHs are responsible for periodically updating and transmitting new prediction model parameters and error acceptance levels to their sensor nodes, as shown in Figure 3. In Li et al. [2009], CHs are also responsible for ensuring that sensor nodes have not stopped working, which may represent, in real deployments, a significant increase in the number of transmissions, besides overloading the network in the case of dynamic scenarios or narrow tolerance for errors.

4.1.1. Decisions in CHs. The Dual Kalman Filter (DKF—presented in Jain et al. [2004]) uses the spatial correlation between measurements from different sensor nodes. Then, the algorithm assesses to the CHs the ability (and the responsibility) to compute several prediction models for each sensor node and choose the best one according to the measurements received. Predictions are made using a modified (distributed) version of the Kalman Filter that takes into account all information that CHs may have, especially from other sensor nodes. Similarly, the Efficient Data Gathering in Sensor Networks (EDGES—presented in Min and Chung [2010]) incorporates not only spatial but also temporal correlations between received measurements.

4.1.2. Decisions in Sensor Nodes. The mechanism proposed in Kho et al. [2009] gives to sensor nodes the ability to make decisions locally using GP regression. There, each sensor node needs to predict the information that is going to be sampled and adjusts its sampling schedule, according to the energy constraints, in order to maximize the information that it will collect during a particular time interval. Simulation results over the data collected from real sensors showed that it is possible to maximize the quality of the information produced by the WSN and reach a high level of confidence by sampling as often as possible, with the constraints imposed on the limited available power in the sensor nodes.

4.1.3. Reducing the Transmissions Between CHs and GWs. The prediction-based monitoring (PREMON) [Goel and Imielinski 2001] exploits spatiotemporal correlations between measurements from different sensor nodes to predict readings that would be done by some sensors and to reduce the number of transmissions done by sensor nodes to their CHs in a DPS. Furthermore, to reduce the number of transmissions between CHs and GWs, CHs may transmit only the prediction model and the updates to the GW, instead of the (aggregated) data that is usually transmitted. Given that GWs and CHs often have higher computational power availability, this kind of architecture supports more robust prediction methods and machine-learning techniques. For instance, Wu et al. [2016] used PCA to reduce the number of dimensions of the data transmitted from CHs to GWs. The main drawback is that the data in the CHs is approximated; that is, it may contain (small) errors introduced by the predictions used to reduce transmissions from sensor nodes, and these errors will propagate to the GWs. To avoid risks of error propagation, it is possible to adopt a more conservative scheme, such as the mechanism called Ken (presented in Chu et al. [2006]). Ken differs from the

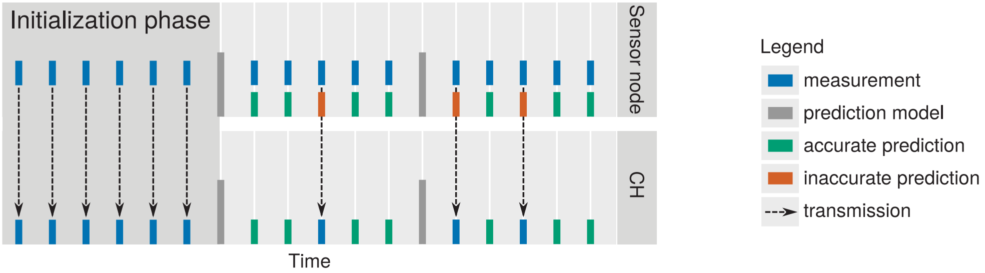


Fig. 5. A variant of the DPS with independent model generation. It is not necessary for any communication between the sensor node and the CH to compute the same prediction model because they are programmed to use the same data.

previous methods because the number of transmissions made by sensor nodes to CHs is not reduced. Instead, predictions are done in CHs and in GWs, and values aggregated in CHs are transmitted if predictions are inaccurate.

4.2. Independent Model Generation

An independent model generation relies on an "initialization phase," that is, a period during which sensor nodes report all the data that they have generated to CHs [Santini and Römer 2006]. The initialization phase ensures that CHs will have complete information about the environment before any prediction model is generated. After the initialization, CHs are able to generate the same prediction models generated in their sensor nodes without making any extra transmission. At this moment, both start predicting the values, with the advantage that the sensor nodes are able to locally verify if a prediction is inaccurate and transmit the actual measurement if needed. Hence, sensor nodes may either regularly report the data to their CHs due to the lack of accuracy in predictions or not report any sensor reading at all, in case that the predictions are sufficiently accurate. Figure 5 illustrates the sensor nodes' and the CH's behaviors.

In Santini and Römer [2006] and Wu et al. [2016], the least mean squares method was used to predict future measurements, which was extended and improved in the simulations made in Stojkoska et al. [2011]. Meanwhile, Debono and Borg [2008] and Aderohunmu et al. [2013a, 2013b] showed results of an implementation using real sensor nodes. Especially, Aderohunmu et al. [2013a, 2013b] compared the savings using several prediction methods: the constant method, weighted MAs, ARIMAs, and the ES. According to their results, the constant prediction method provided the best tradeoff between accuracy and energy consumption in sensor nodes.

4.2.1. Decisions in CHs. According to the mechanism proposed in Jiang et al. [2011], CHs can adapt the sensor nodes' operation according to the potential savings that predictions may introduce. To decide it, the authors use a formula to calculate whether it is worth making predictions in sensor nodes or not, based on the relation between the predictions' accuracy, the correlation between measurements, and the error tolerated by the user. According to the estimated gains, sensor nodes can be set to (i) go to sleep mode, without making any measurement; (ii) make measurements and transmit every measurement done; or (iii) make measurements, transmit them to the CH whenever the prediction differs by more than an accepted value, and update the prediction model parameters when necessary.

4.2.2. Decisions in the Sensor Nodes. In Marbini and Sacks [2004], Ragoler et al. [2004], and Jain and Chang [2004], sensor nodes may have the ability to make further decisions based on the predictions' accuracy. In Ragoler et al. [2004], sensor nodes can decide to

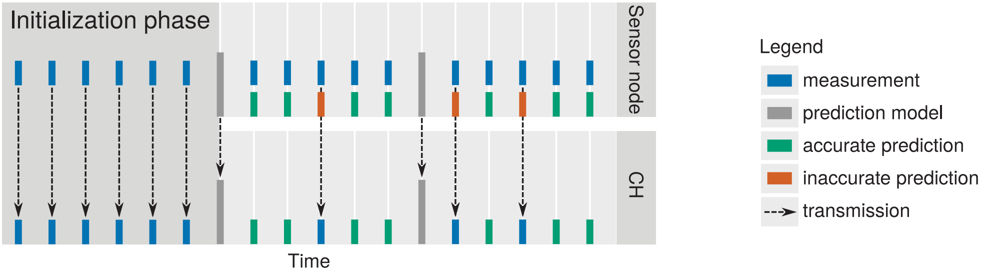


Fig. 6. As sensor nodes can overhear their neighbors' data without overloading the network or congesting the medium, they may locally decide the best prediction method and later inform the computed model to the CHs.

aggregate the data received from their neighbors by suppressing measurements that are inside their confidence interval, instead of forwarding them to the CHs. To make such savings possible, CHs must make the same predictions in order to answer user queries locally. In Jain and Chang [2004], sensor nodes may locally decide whether to adjust their sampling rate or not, based on the accuracy of their predictions. In theory, while local predictions are accurate, their sampling rate can be reduced, and their energy can be saved by turning off their components for longer intervals. In this case, CHs are responsible for controlling the bandwidth consumption to avoid peaks of packet transmissions and, consequently, collisions.

The works done in Raza et al. [2012, 2015] propose a new naive algorithm for predicting. It is a linear approximation that uses recent measurements to calculate the slope of the measurements' trend. Predictions can be easily calculated by sensor nodes using interpolation. According to their simulation results, the suppression ratio can reach high levels (up to 99% of the application data) if the data has low variability. In a realworld testbed, the energy savings were significant (nearly 85% of reduction), even though they did not reach the suppression ratio levels that the simulations had suggested.

4.3. Model Generation in Sensor Nodes

The main drawback of generating prediction models independently was addressed in Le Borgne et al. [2007]: approaches that rely on predefined prediction methods can lead to poor prediction performances if the model choice is not accurately done. The authors decided to generate prediction models in sensor nodes (and not in CHs, the alternative solution), as shown in Figure 6. As in the other DPSs, sensor nodes start transmitting all the measurements to their respective CHs. However, a new responsibility is assigned to sensor nodes: after collecting local measurements, they must fit a prediction model to the real data and communicate any occasional change to their CHs. Fitting a prediction model means finding the model that best summarizes the real measurements. Hence, this mechanism requires much more computing power from the nodes than the other approaches (both to store more data and to choose the prediction models) and the savings depend on the predictions' accuracy, which may vary according to the sensed phenomenon and the data sampling rate. Moreover, the choice of the prediction method is restricted by the memory and process power limitations of the sensor nodes.

Le Borgne et al. [2007] tested how better AR models can improve the WSN lifetime in comparison with constant prediction models. The results of the simulations using real data from WSNs showed that the adaptive approach could reduce the number of data transmissions with neither exceeding the constrained memory nor the computational resources of common wireless sensor nodes. With identical architectures, Li and Wang

[2013] chose the traditional ARIMA and McCorrie et al. [2015] the ES method to predict temperatures in the environment and in aircraft engines, respectively. The hybrid model presented by Askari Moghadam and Keshmirpour [2011] improved the quality of the predictions, giving to sensor nodes the autonomy to adopt an ANN when the predictions using ARIMA were inaccurate. In the worst case, if the predictions using the ANN model also fall outside the accepted threshold, sensor nodes are responsible for transmitting the real measurements and the new models' parameters to their CHs.

Additionally to the predictions, Lattanzi et al. [2013] and Bogliolo et al. [2014] adopted a solution at the hardware level (called wake-up receiver) to reduce the energy consumption during idle periods, improving the communication between sensor nodes. Simulation results showed that the combination of both mechanisms could lead to larger gains than adopting each technique in isolation. The main drawback is that the WSNs' topology cannot change when hardware solutions are adopted because these approaches use directional capsules and might stop working if the sensor nodes move.

4.3.1. Distributed Knowledge. Some works adopt the strategy of disseminating the sensor nodes' knowledge to their neighbors. On the one hand, having more information about the surroundings gives to sensor nodes the ability to compute better prediction models before transmitting their parameters to the CHs. Therefore, more accurate predictions reduce the number of long-distance transmissions between sensor nodes and CHs that usually represent a reasonable overhead in the overall number of transmissions [Lee and Xu 2003]. On the other hand, extra information sharing may represent a waste of resources, because some sensor nodes will never be able to track an object (or a phenomenon) due to their distance or because the extra information is simply not sufficient to improve their predictions.

The dual prediction-based reporting mechanism (DPR) was first introduced in Lee and Xu [2003] and further explored in Xu et al. [2004a], using a prediction-based approach for performing energy-efficient reporting in object tracking sensor networks. The data received from neighbors can be used by a sensor node to predict if an object will be in its range and activate its sensors only if necessary. If its sensors are activated, a sensor node can verify whether its predictions about the object position were accurate and notify the CH about occasional inaccuracies. For the simulations, some prediction methods, such as the constant, the MA, and the ES method, were adapted to the sensor nodes' limitations to reduce computational costs. The simulation results showed that the algorithm is mainly affected by the reporting frequency, but the dynamics of the objects' movements are also important and reflect on the energy saved by these approaches.

The approach in Guestrin et al. [2004] was designed for WSNs that monitor environmental parameters. It relies on the correlation between measurements done by sensor nodes that are closely placed, which requires their exact localization. First, the distributed mechanism creates a junction tree (explained in Paskin and Lawrence [2003]) layer on the top of the routing layer to disseminate the prediction method. During the *dissemination phase*, sensor nodes receive from CHs some information according to their physical location, which may be used to calculate the prediction model parameters. Thus, each sensor node is responsible for calculating part of a linear system using the available information at the time and broadcasting the (partial) results to its neighbors. This is done until every node gets access to the complete information from all other nodes in the same cluster. Once such information is completely disseminated, the parameters of the prediction models are transmitted to the CHs, which makes them able to predict new measurements and reduces the number of transmissions from sensor nodes. Finally, a sensor node updates the prediction model parameters whenever it detects that the difference between the predicted value and the measurement is greater than a fixed threshold. Simulation results using static sensor nodes showed that this mechanism was able to reduce the number of transmissions and save energy, also verifying that it is highly scalable. The authors suggest that it can be extended to make WSNs support hundreds of nodes and run different applications, such as detections of outliers, data compression, and adaptive data modeling. Moreover, even though Guestrin et al. [2004] adopted the kernel regression to predict new measurements, they argued that the distributed mechanism may be extended to support other techniques.

Garrido-Castellano and Murillo-Fuentes [2015] implemented in real wireless sensor nodes the distributed kernel least squares regression method (explained in Predd et al. [2009]). There, each sensor could predict the temperature measurements of its neighbors based on their position and the local measurements. Differently from the mechanism presented in Guestrin et al. [2004], the authors focused on the prediction algorithm and explained its utility in scenarios where sensor nodes must take local decisions based on complex and nonlinearly distributed data. Although the implementation has worked and the algorithm showed the same precision as in the simulation results, the authors listed several problems to deploy the mechanism in a real WSN. Among other problems encountered, they gave special attention to sensor nodes' small memory size and their restricted computing capabilities, which required a workaround to handle floating-point operations.

The approach presented in Carvalho et al. [2011] also distributes tasks among sensor nodes. However, instead of requiring the exact position of each sensor node, they rely on their proximity to find correlated measurements. Each node is responsible for calculating a linear regression based on its different measurement types. For instance, a sensor node equipped with temperature and relative humidity sensors may use measurements of temperature to predict the relative humidity at a certain time. After calculating the regression coefficients, each node broadcasts them to its neighbors. Every neighbor that observes the same coefficients may inform the CH that its measurements are similar. CHs receive coefficients and predict future measurements and, from this moment, a sensor node will only transmit the measurements if the predictions are inaccurate. The simulation results showed an improvement in the energy consumption and a reduction in the number of transmissions, compared with the default operation. However, the authors did not take into consideration the energy consumption to calculate the coefficients of the linear regression, nor the extra space used to store the measurements for the regression analysis.

4.3.2. Autonomous Sensor Nodes. Some authors propose that sensor nodes should have complete autonomy and decide for themselves, based on predictions, if they should pause on making measurements for a period. Shen and Li [2008] developed a way to calculate whether sensor nodes should turn some of their components off or not, based on their total amount of energy and the time required to switch off their components. Moreover, each sensor node is able to calculate the possible amount of energy saved and to decide when to change its internal status. That is, based on the predictions, a sensor node only changes its internal status if some energy is going to be saved. For the predictions, they use Wavelet Neural Networks—an extension of ANNs.

5. COMPARISON OF THE PREDICTION METHODS ADOPTED IN WSNS

In the previous sections, we presented and discussed the architectures and schemes that support the adoption of prediction methods in WSNs. Moreover, we highlighted the degree of autonomy that CHs and sensor nodes may have in some cases, which impacts the decision about the prediction method used. For instance, a scheme that forces sensor nodes to work autonomously may not achieve the desired gains if a computationally

Table I. List of Time Series Methods

Method	Preprocessing Time Complexity	Runtime Complexity	Space Complexity	Reference in WSNs
Constant prediction	None	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Goel and Imielinski [2001], Lee and Xu [2003], Xu et al. [2004b], Aderohunmu et al. [2013b], Emekci et al. [2004], Ragoler et al. [2004], Marbini and Sacks [2004], Jain et al. [2004], Jain and Chang [2004], Aderohunmu et al. [2013a], and Bogliolo et al. [2014]
AR(p)	$\mathcal{O}(k\ h^2)$	$\mathcal{O}(p\ w)$	$\mathcal{O}(p)$	Tulone and Madden [2006], Jiang et al. [2011], Le Borgne et al. [2007], and Aderohunmu et al. [2013a]
MA(q)	$\mathcal{O}(k\ h^2)$	$\mathcal{O}(qw)$	$\mathcal{O}(q)$	Xu et al. [2004b] and Aderohunmu et al. [2013a]
Exponential Smoothing (ES)	$\mathcal{O}(k^3\ h)$	$\mathcal{O}(w)$	$\mathcal{O}(1)$	Xu and Wolfson [2003], Xu et al. [2004b], Aderohunmu et al. [2013a], and McCorrie et al. [2015]
ARIMA(p, d, q)	$\mathcal{O}(k^3h^2)$	$\mathcal{O}((p+q)w)$	$\mathcal{O}(max(p, q+1))$	Li et al. [2009], Li and Wang [2013], Aderohunmu et al. [2013b], Liu et al. [2005], and Askari Moghadam and Keshmirpour [2011]

intensive prediction algorithm is adopted. On the other hand, a scheme that exploits the extended computational power of the CHs would be subutilized if a simplistic (so-called *naive*) method was adopted.

Therefore, before discussing the aspects that should be taken into consideration before adopting a prediction-based data reduction scheme in a WSN (in the next sections), we present the prediction algorithms that have been used in WSNs. At first, we split them into three classes: time series methods, regression methods, and machine-learning techniques. For each algorithm, we highlight its potential to reduce the number of transmissions according to the data characteristics, before discussing other particularities that may impact its adoption in WSNs, such as time and space complexities.

5.1. Time Series Methods

A time series is a sequence of data points, typically consisting of observations made over a time interval and ordered in time [Box et al. 2008]. Each observation is usually represented as x_t, where the observed value x is indexed by the time t in which it was made. Thus, a *time series prediction model* uses a time series as input to make predictions. These predictions are represented as a function of the past observations and their respective time, that is, $x_t = f(x_{t-1}, \ldots, x_{t-k})$, where the function f is a prediction model usually defined by parameters calculated using past observations. The algorithms used to find acceptable values for the parameters (i.e., those that may generate accurate predictions) may require some extra computation before making any prediction. Furthermore, since the environment may evolve and change, the parameters used to define a model may become obsolete after a while and hence the predictions' accuracy may decrease. Therefore, there is a computational cost to update the parameters of the chosen method and keep the predictions' accuracy.

In the following, we explain the time series methods (also called *forecasting methods*) used in WSN environments: the naive approaches, the Autoregressive (AR), the Moving Average (MA), the Exponential Smoothing (ES), and the Autoregressive Integrated Moving Average (ARIMA). Table I summarizes their characteristics and the reader can refer to Makridakis et al. [1998], Box et al. [2008], and Hyndman and Athanasopoulos [2014] for detailed information on forecasting methods.

Pros. The main advantage of time series methods is their independence. That is, it is not necessary neither external data nor an extended data analysis to make accurate predictions. Moreover, each wireless sensor node may be able to predict short time intervals without depending on the support from neighbors' or CHs' computing power, due to the low space-time complexity of such methods.

Cons. Most of the time series methods adopted in WSNs assume that there is only one data type and neglect the presence of multiple sensors in a node, which is not uncommon and may be better exploited by other methods. Furthermore, their accuracy is (usually) significantly lower when they are used for long-term predictions. That is, the data usually has similar values in the near future, but unobserved phenomena and sensor nodes' constraints may have a long-term impact, affect the predictions' accuracy, and lead to fundamental errors, such as negative dimensions.

5.1.1. Naive Approaches. Examples of naive approaches for predictions may vary between the *average* of the past observations, the *maximum* observed value, or exactly the *same* as the last observation made in time. Thanks to their simplicity, even though the predictions are not the most accurate as they could be, naive approaches are usually compelling options for WSNs composed by sensor nodes with energy constraints or low computing power.

Pros. There is no complex data processing in naive approaches, which makes them an affordable option in terms of computing time and required memory space. Typically, naive approaches assume that future values can always be computed in constant time, given the historical data.

Cons. If the data has high variability, these methods are usually more inaccurate and imprecise than the most advanced methods. In other words, if the tolerated error threshold is small, they will fail most of the time. Otherwise, they will fail less often, but their precision will also be reduced because they will consider a wider range of values to assess predictions as accurate.

5.1.2. Autoregressive (AR). Regressive models are used to represent the expected value of a variable given the values from a set of correlated variables. In an autoregressive (AR) model, the mean and the variance of the values are constant, and the estimation is a linear combination of the values from the same variable. An AR model with order p (referenced as AR(p)) is defined by a set of p coefficients $\alpha_1, \ldots, \alpha_p$. The parameter p is usually set among k possible values, using an information criterion measure (usually, $k \in [0, 6]$). Then, a value observed at time t can be represented as

$$x_t = c + \sum_{i=1}^{p} \alpha_i x_{t-i} + \varepsilon_t, \tag{1}$$

where the term ε_t is a Gaussian white noise with variance σ^2, and c is a constant such that, in the case of having a stationary process with mean μ,

$$c = \mu \left(1 - \sum_{i=1}^{p} \alpha_i\right). \tag{2}$$

The value $\hat{x}_{t+1}$ can be predicted as $\hat{x}_{t+1} = c + \sum_{i=1}^{p} \alpha_i x_{t+1-i}$ using the same set of parameters α_i.

Pros. A trend observed in the (relatively) near past can be used to improve the predictions about the near future. Moreover, the AR method is very efficient for short-term predictions for two reasons: (i) it is less sensitive than the naive predictions against the data noise, and (ii) its predictions follow trends observed in the most recent observations, which represents a potential to keep a high accuracy even if the data has high variance.

Cons. Long-term predictions using AR tend to be inaccurate, due to uncertainty about the order of the model, its coefficients, and unobserved errors. Especially when the predicted period is longer than the order of the model, predicted values are used for making new predictions, which propagates errors and affects the overall accuracy.

5.1.3. Moving Average (MA). Similarly to the AR models, an MA model is defined by an order q and is referenced as MA(q). Its order is defined as a window length q that represents the number of past measurements that will be taken into account in the predictions. As in the AR method, the value of q is usually set among k possible values using an information criterion measure (usually, $k \in [0, 6]$).

During the learning phase, which takes at least q time intervals, the algorithm stores the measured data that will be used to calculate the model parameters and make the predictions. In order to incorporate eventual changes in the future predictions, the MA method calculates the weighted average of the observations recently made. The prediction of the value of x at time t is calculated using the following formula:

$$x_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}, \tag{3}$$

where μ is the average of the last q values,

$$\mu = \sum_{i=1}^{q} \frac{x_{t-i}}{q}, \tag{4}$$

$\{\theta_i \mid i \in 1, \ldots, q\}$ are parameters of the model, and $\{\varepsilon_i \mid i \in 1, \ldots, q\}$ are the white noise error terms, which $\varepsilon_i = x_i - \mu$.

Pros. Using the MA method, events that influenced a value observed at time t can only influence the most recent observations; that is, the predictions do not follow short-term trends. Moreover, this method can also be used to remove casual noise from the data, given that its predictions are less sensitive against outliers than those made using AR models, for example.

Cons. In order to calculate the parameters of the model (θ_i), it is not possible to use linear least squares, and iterative nonlinear fitting procedures are required, which makes the MA method computationally more complex than the AR one. Furthermore, similarly to AR, the accuracy of this method significantly decreases when predicting more values than its own order. This occurs because, after the q^{th} prediction, it does not have any actual observation to compare and the predictions tend to an average value that usually does not match with the real observations.

5.1.4. Exponential Smoothing (ES). The simplest version of the ES is also known as Exponentially Weighted Moving Average (EWMA), and the value predicted for the time $t + i$ can be calculated using only the most recent observation and the most recent forecast. For instance, the value of $\hat{x}_t$ is the weighted average:

$$\hat{x}_t = \alpha x_{t-1} + (1 - \alpha)\hat{x}_{t-1}. \tag{5}$$

Guided by the value of $\alpha \in [0, 1]$ (also called *smoothing constant*), the relevance of the old measurements undergo an exponential decay, which justifies its name.

Other formats of the ES are also widely used, adding up to two new parameters (β and γ) in order to better detect nonlinear trends. A common way to set up good values for α, β, and γ is by trying among k possible values each (usually, $k = 10$). The choice is made according to the errors observed over the data already observed, for example, calculating a prediction $\hat{x}_{t-1}$ and comparing it with the real observation x_{t-1}.

Pros. Space and time complexities are smaller when compared with the AR and MA methods, and the predictions incorporate better the trends in the last observed values.

Cons. It has some of the same limitations observed in the MA methods, such as the low efficiency when predicting the data value even in short time intervals. Moreover, its confidence intervals increase exponentially.

5.1.5. Autoregressive Integrated Moving Average (ARIMA). An ARIMA model defines a stationary process that is composed of the combination of an AR and an MA model. Values that will be observed in the future can be more accurately predicted if the calculation considers (i) the magnitude of the last observations and their trends (incorporated by the AR model) and (ii) the impact of (unobserved) shocks that influenced their current state (incorporated by the MA model). In the case of having nonstationary data, an initial differencing step (corresponding to the "integrated" part of the model) can be applied. Such a transformation can be represented by the following equation:

$$y_t = (1 - L)^d x_t, \tag{6}$$

where d is the order of the integrated model and L is the Lag operator, such that $L^k x_t = x_{t-k}$ for all $t > k$.

An ARIMA(p, d, q) model contains an AR model with order p and an MA model with order q and a value observed at time t can be represented as

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}. \tag{7}$$

Note that this formula is used to predict the value of y_t, which is derived from x_t, if $d > 0$, or simply equal to x_t, if $d = 0$. Moreover, as well as the AR, MA, and ES methods, the parameters p, q, and d are usually set among k possible values each, using an information criterion measure (usually, $k \in [0, 6]$).

Pros. The ARIMA method has higher accuracy than the previous methods, given that the predictions consider new trends observed in the latest observations and converge much slower than the MA models to the average values. Furthermore, this method has a particular characteristic, which is the overlapping with other methods. That is, some ARIMA models are equivalent to other methods, for example:

- ARIMA(0, 1, 0) is equivalent to the naive method that assumes the last observation will repeat in the future.
- ARIMA(0, 2, 0) is equivalent to the naive method that assumes a linear increasing based on the last two observations.
- ARIMA(0, 1, 1) is the simplest model of the ES (with only one parameter).
- ARIMA(0, 2, 2), ARIMA(0, 1, 2), and ARIMA(1, 1, 2) are equivalent to more complex ES models.

Hence, when choosing the ARIMA model that fits more properly to the data, some of the other methods are also implicitly considered.

Cons. The time and space complexities are bounded by the worst complexity between the AR and the MA methods, which depends on the values of p and q (as shown in Table I). Moreover, an extra step may be required to differentiate the data.

Table II. List of Regression Methods

Method	Preprocessing Time Complexity	Runtime Complexity	Space Complexity	Reference in WSNs
Linear regression	$\mathcal{O}(nd)$	$\mathcal{O}(2^{n-\bar{n}})$	$\mathcal{O}(d^2)$	Yann-Ael and Gianluca [2005], Deshpandem et al. [2004], Chu et al. [2006], Debono and Borg [2008], Min and Chung [2010], Matos et al. [2010], Carvalho et al. [2011], and Wu et al. [2016]
Multivariate Kernel regression	$\mathcal{O}(n^3d^3)$	$\mathcal{O}(d)$	$\mathcal{O}(nd^2)$	Guestrin et al. [2004] and Garrido-Castellano and Murillo-Fuentes [2015]
Principal Component Analysis	$\mathcal{O}(p^2m + p^3)$	$\mathcal{O}(p^{2.3})$	$\mathcal{O}(p^2)$	Guestrin et al. [2004]
Gaussian Process regression	None	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	Kho et al. [2009]

5.2. Regression Methods

Regression methods have a different approach than time series methods. Instead of relying only on past values to make predictions, they also predict measurements based on other measurement types. For instance, given a value observed by one sensor node, a regressive model can be used to predict which value would be observed by another sensor node. In the following, we discuss three methods used in WSN environments: the linear regression, the kernel regression, and the PCA. Table II summarizes their characteristics and the references that applied each of them.

Pros. It is possible to combine different data types, for example, to use temperature measurements to predict the relative humidity at the same time or some moments later. In other words, regression methods consider that the environment is composed of more than a single type of data and that other factors may influence the studied process, which fits the distributed architecture of WSNs.

Cons. In comparison with time series methods, regression methods have higher space and time complexities that, given the constrained capacities of some sensor nodes, may limit their adoption in WSNs.

5.2.1. Linear Regression. Linear regressions are the simplest kind of regression. They are used to characterize linear relations between the observed variables. Using linear regression, it is possible to use a measurement x to predict the value of y based on a linear function $y = \beta_0 + \beta_1 x$. The coefficients β_0 and β_1 can be calculated using the least squares method (see Diez et al. [2012] for other methods).

Pros. It is possible to make three different types of predictions that are useful in WSN scenarios [Deshpandem et al. 2004]: (i) range based, that is, to predict whether a measurement will be inside a range of values; (ii) value based, that is, to calculate the probability that a measurement will be a certain value; and (iii) average aggregation, that is, to predict the average of a set of unobserved measurements at a time.

Cons. Linear regressions assume normally distributed data, which may rarely happen when considering several sources of data and requires within-study correlation estimates.

5.2.2. Kernel Regression. Kernel density estimation is a nonparametric model used to estimate the probability density function of the observed data, starting with no assumptions about the data distribution. The goal of the kernel regression is to find the value of $E[Y|X] = m(X)$ for an unknown function $m(\cdot)$. To achieve that, a regression is made based on the given values of X, with the help of a kernel function that is responsible for quantifying the similarity between their data points. Finally, a new probability density function is drawn based on the observed values and can be used to predict the value of $E[Y|X]$.

Pros. It does not require any assumption about the data distribution and tends to have smaller errors when compared with the linear models.

Cons. To compensate for the absence of assumptions, more data is required to find a proper approximation to the real distribution. Hence, its computational complexity is much higher than that of linear regressions, in terms of both space and time.

5.2.3. Principal Component Analysis. The PCA method is used to reduce the dimensionality of datasets. It uses orthogonal transformations to convert sets of observations of (possibly) correlated variables into sets of values of linearly uncorrelated variables, so-called *principal components*. After the conversion, only the principal components that retain most of the variation present in the dataset are kept. Thus, based on the information retained by these components, it is possible to predict the values of the original dataset with a high degree of confidence. For instance, it may be possible to predict the measurements of all the sensor nodes based on the measurements made by those that observe the highest variations in the environment. The algorithm to calculate the principal components uses eigenvectors and a covariance matrix and is thoroughly explained in Jolliffe [2002].

Pros. The PCA method provides a means to remove part of the data without losing the relevant information that it contains.

Cons. The preprocessing phase involves the computation of the product of matrices based on large sets of data. It means higher time and space complexities than the other options.

5.2.4. Gaussian Process Regression. In short, a Gaussian Process (GP) is a collection of random variables. There, a finite set of such random variables has a joint multivariate Gaussian distribution; that is, it is defined by their means and the covariance of the distributions. Each random variable ($f(x)$) is indexed by x and a covariance function that incorporates prior assumptions about the relation with the other distributions. Because of that, no advanced knowledge about the data is required for making accurate predictions. In Rasmussen and Williams [2006], the GP regression method is explained in detail.

Pros. The GP regression method produces probabilistic models that are composed of single values and associated confidence intervals, expanding their possible applications. Furthermore, the GP regression is well known because of its higher accuracy when compared to other regression methods, thanks to its fast fitting to the underlying (unknown) data distribution.

Cons. The main drawback of the GP regression is the computation time required to make a prediction. As is shown in Table II, the computation has a cubic growth and new models cannot be generated online, which makes it an unfeasible option for larger datasets. It is worth mentioning that there are works focused on reducing its computational complexity in exchange for reducing the accuracy [Saat 2011].

5.3. Machine-Learning Techniques

As shown in Alsheikh et al. [2014], machine-learning techniques have been adopted in several solutions for WSNs at different levels, such as routing, medium access control, and event detection. However, from these solutions, only Artificial Neural Networks (ANNs) have been applied for reducing their number of transmissions.

5.3.1. Artificial Neural Networks. The methods described in the previous sections are mainly considered *traditional methods*, given that they are extensions of probabilistic approaches. As described in Jain et al. [1996] and Haykin [2009], ANNs are based on a different paradigm and their general idea is to create an artificial version of the biological neurons, that is, to simulate a network of components (so-called *neurons*) and predict a system's output, given a set of inputs. To achieve that, an ANN must go through a *learning phase*, that is, to adapt its internal parameters and learn from available historical data.

Pros. The main advantage of ANNs is that they are able to handle multiple data types and model regressions between several variables. Additionally, ANNs have been observed to perform more accurately than the traditional methods in time series with discontinuities, which may happen in the case of absence of parts of the data—very common in some WSNs. Finally, Kang [1992] found that ANNs often perform more accurately for long-term predictions than for smaller intervals.

Cons. ANNs are soft computing solutions that cannot be bounded by a computational time limit. One of the reasons for the high computational costs is that the design process of an ANN involves (1) selecting the number of hidden layers, (2) adjusting the connection between each layer (the synapses), (3) choosing the number of neurons in each layer, and (4) setting the activation function, a learning algorithm, and the number of training samples.

ANNs are powerful and can approximate other prediction methods and nonlinear models, given the best conditions, such as enough data. However, the amount of data required to find a stable ANN is much higher than the others, because it has more parameters to estimate. Moreover, there is no theoretical guarantee that they will perform well for out-of-sample forecasts, that is, predictions after the learning phase. In conclusion, it may be costly to find out which situations better fit neural networks than traditional models.

6. DISCUSSION

In this section, we answer some questions that may arise during the design of a strategy to adopt predictions to reduce the number of transmissions in a WSN:

(1) How to improve a WSN using predictions?
(2) How to choose a prediction model?
(3) Why (not) make predictions in GWs?
(4) Why (not) make predictions in sensor nodes?
(5) Why (not) make predictions in CHs?

The discussion is supported by the statistical literature (partially represented by the prediction methods explained in Section 5) and by the results presented in the works included in this survey.

6.1. How to Improve a WSN Using Predictions?

Some characteristics of a WSN are less flexible than others. For example, a WSN that was designed and deployed to track objects cannot be simply changed to monitor room temperature, because its sensors measure other parameters than temperature, relative humidity, solar radiation, or any other value that could be correlated with the local temperature. Therefore, the WSN type is a firm characteristic, because it would be highly costly to change sensors in nodes already deployed.

On the other hand, one may have some information about the data that is going to be monitored by a WSN at the moment of the deployment, but there are inherent costs to acquire real data from the environment and perform its analysis. Such costs must be taken into account when calculating the improvements brought by the use of predictions, but they are not as impeding as changing the WSN type.

Therefore, to answer "how to improve a WSN using predictions?" we describe a bottom-up approach that goes from less flexible characteristics, such as the types of sensors available in a WSN, to the economically cheapest and the least time-consuming ones.

6.1.1. WSN Type. There are three large classes of WSNs: one encompasses the networks used for event detection; another the WSNs for monitoring and reporting (query based or continuously); and the networks used for objects (and people) tracking. Each of these classes has different requirements about timeliness, throughput rate, computing power in sensor nodes, and tolerance to packet losses. Hence, for instance, it is unlikely to reutilize in a fire detection scenario a WSN that was originally deployed to monitor the temperature, due to the economic and death risks that it would involve. In other cases, it may be nearly impossible to adapt one WSN to make another task, because their sensor nodes may not contain the proper hardware to measure a certain parameter, as explained before.

Therefore, the WSN type must be the first aspect to be taken into account when considering the use of predictions to reduce the number of transmissions in a WSN. In the following, we list the requirements of each WSN type and give examples of applications that can be used in each case. As a reference, in Table III, we filled the second column with a letter **E** to identify prediction-based solutions adopted in event detection applications, **Q** in query-based solutions, **C** for continuous monitoring, and **O** for object tracking ones.

Event Detection. Such networks are very strict about the delays that the transmissions may suffer because having an undesired delay when reporting an event may cause economic losses or put lives at risk, depending on the situation (e.g., in a disaster detection). However, if all sensor nodes transmit at the same time, it may cause congestion. As a consequence, the number of packet collisions may increase and delay the delivery of relevant information. Therefore, these WSNs should avoid approaches that ignore the delay of the packets as a critical issue, due to the risks that it can bring to the environment surrounding the WSN. It is also possible to adopt regression methods or machine-learning techniques to predict events, which requires advanced knowledge about the domain under study and specific refinements according to the scenario.

Query-Based Monitoring. Monitoring WSNs are used to measure environmental parameters, people's health, machines, and engineering structures with low computational power and energy supplies. Especially, query-based monitoring WSNs work in a pull-based fashion; that is, users eventually demand information from the environment and the CHs make an effort to answer most of the queries locally and transmit them as rarely as possible to sensor nodes. As in most cases user queries contain which information must be retrieved and the error tolerated by the user, the CHs may predict the sensors' measurements and avoid transmissions whenever the confidence levels of the predictions match the users' expectations.

Continuous Monitoring. As in query-based WSNs, it is common to encounter temperature, relative humidity, light, solar radiation, wind speed, and soil moisture sensors, among others, that can measure environmental parameters. The main difference may be in the density of sensor nodes in a deployed network. If the WSN is set to continuously transmit the state of the environment, some sensor nodes may run out of

Table III. Characteristics of the Approaches Considered in This Article

Reference	WSN and Data Type	Does In-Code Computation	Does Extra Transmissions	Turns the Nodes Completely Off	Requires Knowledge About Location	Requires Knowledge About the Data
Goel and Imielinski [2001]	O	✓	✓		✓	
Cheng et al. [2003]	Q			✓	✓	
Lee and Xu [2003] and Xu et al. [2004a]	O	✓	✓		✓	✓
Lazaridis and Mehrotra [2003]	C	✓	✓			✓
Marbini and Sacks [2004]	O	✓				
Emekci et al. [2004]	Q			✓		✓
Deshpandem et al. [2004]	O			✓	✓✓	✓
Jain et al. [2004]	O	✓	✓		✓	✓
Guestrin et al. [2004]	C	✓	✓		✓✓	✓
Xu et al. [2004b]	O	✓	✓	✓	✓✓	
Ragoler et al. [2004]	C	✓		✓		
Jain and Chang [2004]	C	✓	✓	✓	✓✓	✓
Yann-Ael and Gianluca [2005]	Q			✓	✓	
Liu et al. [2005]	C	✓	✓			
Chu et al. [2006]	C	✓	✓			✓
Santini and Römer [2006], Stojkoska et al. [2011], and Aderohunmu et al. [2013a, 2013b]	C	✓				
Tulone and Madden [2006]	Q	✓	✓			✓
Le Borgne et al. [2007]	C	✓	✓			
Shen and Li [2008]	C	✓		✓		
Debono and Borg [2008]	C	✓				✓
Li et al. [2009]	Q	✓	✓			✓
Kho et al. [2009]	C	✓	✓	✓		
Min and Chung [2010]	C	✓	✓		✓	✓
Matos et al. [2010]	C	✓	✓			
Jiang et al. [2011]	C	✓	✓	✓	✓	✓
Samarah et al. [2011]	O	✓	✓	✓	✓✓	
Malik et al. [2011]	Q			✓		✓
Askari Moghadam and Keshmirpour [2011]	C	✓	✓			
Carvalho et al. [2011]	C	✓	✓			
Li and Wang [2013]	C	✓	✓			
Bogliolo et al. [2014]	C	✓				
Yin et al. [2015]	Q	✓	✓			
Raza et al. [2015]	C	✓	✓			
Garrido-Castellano and Murillo-Fuentes [2015]	C	✓	✓		✓✓	
McCorrie et al. [2015]	C	✓	✓			✓
Wu et al. [2016]	C	✓	✓			

battery quicker, and their data should be instantly replaced by the data collected in the same region. Also, thanks to the data redundancy inherent to densely deployed WSNs, they usually have fewer restrictions about delays that may occur during the transmissions. Hence, adaptive sampling mechanisms can use predictions to avoid unnecessary transmissions and substitute the sensed data that has not been reported to CHs in a DPS.

Object Tracking Sensor Networks (OTSNs). Some WSNs are responsible for tracking objects. Their tasks may vary from simply detecting the presence of a person in a region to applications that track the (almost) exact position of an enemy in a battlefield, an animal in a farm, or cars in a smart city. These WSNs are less tolerant to delays in data delivery and require more detailed information from sensor nodes, such as high-precision measurements [Xu et al. 2004b]. Because of the importance of the detailed information, this kind of application usually requires sensors that consume more energy and are more expensive, such as cameras and microphones.

In order to keep the timeliness in the data delivery, it is important to avoid packet collisions and medium congestion. Thus, transmissions are made by as few sensor nodes as possible at a time, differently from monitoring applications that keep collecting as many parameters as possible to build the most complete visualization of the environment in the GWs. To do that, OTSNs adopt sensor nodes with higher computational resources, which are able to autonomously decide the best times to measure or transmit [Bhatti and Xu 2009]. Hence, the computation in sensor nodes may be heavier, and an SPS with model generation in sensor nodes can help to meet their strict timeliness and reliable data delivery requirements.

6.1.2. Energy Resources. Limited energy resources are one of the main constraints in WSNs. However, some of the works presented in this survey assume that some sensor nodes have larger energy resources than others, which may occur thanks to more powerful (and therefore more expensive) batteries or energy harvesting, or simply because they are plugged in. In conclusion, CHs may exploit the abundance of their energy resources to choose a prediction method that is more complex and probably more accurate than the others.

As shown in Tables I and II, some methods have much higher runtime complexity than others; for example, the GP regression has a cubic growth, while the ES grows linearly. Therefore, if sensor nodes have scarce energy resources, it may be more rational to restrain the options to the least complex methods. Finally, it is worth mentioning that, besides the energy resources, the processing power of the sensor nodes plays an important role in the adoption of prediction methods. We discuss such limitations in the following.

6.1.3. Processor. The computational power of the sensor nodes is determinant to decide which kind of operation they will perform. Sensor nodes with the capacity to compute complex operations can be set to generate a set of prediction models and choose the one that better fits the current data, make predictions, and compare with real measurements. On the other hand, if the sensor nodes' processors cannot perform complex mathematical instructions, they may be limited to naive prediction methods, to predict values using models calculated by CHs, or to simply compare predictions computed by CHs with real measurements.

Furthermore, the relation between the energy consumed when executing a machine instruction and the energy spent to make a radio transmission has been considered in few works, but has been shown as a relevant aspect by some authors. That is, many authors assume that processing data is always less resource wasting than making a radio transmission. However, in Goel and Imielinski [2001], it has been shown that their approach is energy efficient only if a prediction can be computed using fewer than 15.000 machine instructions in their environment. Moreover, given that some sensors have much higher consumption than the others (especially in OTSNs, as explained in Anastasi et al. [2009]), such a number can significantly vary from case to case.

In Table III, we used the third column to show the works that compute extra instructions in sensor nodes, for example, predictions and/or their models. The fourth column shows checkmarks in the works that make extra transmissions, either because of the prediction models' parameters or the decision about adopting predictions, taken in runtime. The fifth column shows the works that completely turn sensor nodes off (i.e., both the micro-controller unit and the embedded sensors), impacting the overall consumption provoked by sensing tasks and transmissions.

6.1.4. Storage Space. WSNs are typically composed by cheap wireless sensor nodes and, in order to keep their costs low, their amount of available memory is extremely limited. In Tables I and II, we show the space complexity to store a prediction model, which can affect the decision about adopting a certain prediction method or not, given the sensor nodes' limitations. Alternatively, the storage space problem may be solved at some cost using external flash memory in sensor nodes (as done in Li et al. [2009]), which significantly increases the number of possibilities for prediction methods and may improve the predictions' accuracy.

6.1.5. Information About the Location of the Nodes. Using current localization techniques for WSNs, it may be possible to calculate the sensor nodes' exact location on the earth based on information retrieved from a Global Positioning System (GPS) device connected to the WSNs [Mao et al. 2007]. Alternative schemes can be used in cases when sensor nodes have no information about their absolute position on the surface of the earth, but they are able to calculate their relative position inside the WSN on different levels of granularity [Pereira et al. 2012]. For example, they may be able to calculate the distance to their closest neighbors or to assess how many hops they are away from the GW.

The level of detail in the information about the sensor nodes' location may lead to decisions about which kind of prediction is going to be made. For example, in the case of having the relative position of the nodes, it may be possible to predict an aggregated function (e.g., the average) of the measurements from sensor nodes placed in a certain region (as done in Cheng et al. [2003]). If no information about their location is available, predictions may be aggregated according to the sensor nodes' data similarity, which can be observed after a training phase (as done in Carvalho et al. [2011]). Moreover, when no information about the location is available, it is possible to opt for a solution in which no aggregation is done, where each sensor node is separately predicted by a different prediction model (as done in Debono and Borg [2008]).

As a reference, we used the sixth column in Table III to illustrate, in each work, how much information about the location of the sensor nodes was necessary. One checkmark is shown if only the relative information was required, for example, the number of hops from the CH to the node or the WSN topology. The presence of two checkmarks means that the exact position of the sensor nodes was used, for example, their GPS position. Finally, in the absence of a checkmark, no knowledge about their localization was used.

6.1.6. Historical Data Availability. Some prediction methods (e.g., ANNs) require large amounts of historical data to generate an accurate prediction model. However, in some scenarios, the data that is going to be measured by the sensor nodes cannot be observed or studied before selecting the best prediction method. This lack of information may reduce the options of possible prediction methods that can be successfully applied. As a reference, we added in the last column of Table III the information about which works assumed a priori knowledge about the data that they will work on.

When no assumptions about the data can be made, their statistical characteristics are not available, or the historical dataset is absent, a "learning phase" may be required. The "learning phase," similar to the "initialization phase" in the DPSs, is a period during which the sensor nodes report all the data that they have generated to the CHs [Santini and Römer 2006]. Adopting a periodic "learning phase" (e.g., once a day) can improve the predictions' accuracy or expose when some prediction models are not performing as accurately as before. Finally, the inherent costs of this procedure must be included in the plan of selecting the most proper prediction method for a specific scenario.

6.2. How to Choose a Prediction Model?

Considering the multiple options to make predictions in different device types, the choice of the prediction model may lead to a successful deployment or simply make it extremely inefficient. Usually, a richer model can provide more accurate results, but it may require more communication among sensor nodes, larger memory buffers, or more computing time. As recommended in Lazaridis and Mehrotra [2003], this choice must be done using experimentation, expert opinion, or past experience to choose between competing models. It has been pointed out in Le Borgne et al. [2007] that "an inadequate a priori choice of a prediction model can lead to poor prediction performances."

In the literature about statistical methods [Timm 2002], a common way to choose a model among a list of options is to reward its accuracy and, in change, penalize its selection according to the number of parameters used to compute the predictions.

Therefore, the first step is to assess the predictions' accuracy. This can be done in several ways by using measures that are supposed to attest the quality of a prediction model in a certain use case. Examples of such measures are the well-known Mean Square Error (MSE), the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE), the Relative MAE (RelMAE), the Mean Absolute Percentage Error (MAPE), and the symmetric Mean Absolute Percentage Error (sMAPE), among others [Hyndman and Koehler 2006]. Once the accuracy has been measured, it is necessary to measure the relative quality of the model. Methods such as Akaike Information Criterion (AIC—presented in Akaike [1974]) and the Bayes Information Criterion (BIC—presented in Schwarz [1978]) are some of the existing options.

In Li and Wang [2013], the methods described previously were used to choose the best ARIMA model to make the predictions in their tests. However, as we described in Section 2, WSNs have computational limitations that most networks do not have. If the chosen architecture requires that the prediction model must be fixed a priori (and cannot be adaptively chosen), the decision must be made based on a few aspects that have an influence on the energy consumption of the sensor nodes, for example, the number of messages generated by the scheme when the prediction fails and all the engineering concerns, such as the energy consumed to (re-)fit prediction models and, especially, to transmit their parameters.

To overcome the limitations of traditional methods, Liu et al. [2005] created a way to select a prediction model that considers the percentage of transmitted measurements (r) and the user-desired level of accuracy (α). Later, Aderohunmu et al. [2013b] designed an extended model for the Prediction Cost (PC), which is more generic and also considers the computational costs of each algorithm in the sensor nodes with respect to their memory footprint (*Ec*):

$$\mathrm{PC} = [\alpha f(e) \ + \ (1-\alpha)r]\mathrm{Ec}, \tag{8}$$

where e is the measure of the predictions' accuracy (e.g., MSE, RMSE, sMAPE) and $f(e)$ is the accuracy according to the chosen measure.

Furthermore, we observed that among aspects that may become extra costs as a consequence of the chosen prediction method, the most important are the computation time required to prepare the model, the required data assumptions, and the computing power, including the extra memory required to make new predictions. As an example, some model parameters can be adapted on the fly by using adaptive filters (e.g., as done in Santini and Römer [2006]), and hence there is no need to store large sets of past data. On the other hand, they may require much extra computation and extra storage capacities from sensor nodes and CHs.

In Tables I and II, we organized the traditional methods described in Section 5 according to their type: time series and regression methods. It is possible to observe that some methods have low space and time complexity (e.g., constant predictions and exponential smoothing). However, some options (e.g., AR, MA, and ARIMA) have similar preprocessing time complexity and the space required by their models, together with their accuracy, should influence the final decision about which one to adopt. The ANN method has not been included in any table because there is no exact solution and no upper limit for its complexity. Usually some heuristics are adopted, for example, the maximum number of runs, but its performance depends on each use case and may be adjusted according to the results obtained.

6.3. Why (Not) Make Predictions in GWs?

As explained in Section 3, GWs can be used to make predictions that are going to be used by the WSNs. The first reason to make predictions in GWs is that they are supposed to be the most powerful device in WSNs (in terms of computing power and energy supply) and, therefore, they may be able to utilize more information about the environment, external changes, and historical data.

6.3.1. Pros. Existing mechanisms show that it is not necessary to establish communication between the GW and a sensor node in order to predict its values; that is, it is possible to predict measurements that a sensor node is going to make based on its neighbors' measurements. As a consequence of this, a sensor node can have its MCU completely turned off for a while, which may be an important step in the direction of improving the overall WSN lifetime. Furthermore, since GWs have access to information retrieved from several locations, they are able to see the broad picture of the environment, better understand how it is evolving, and infer predictive models at a larger time-space scale, accounting for possibly existing cyclic behavior, global trends, or other aspects not discernible from the sensors' limited (time- and space-wise) perspective.

6.3.2. Cons. On the other hand, predicting future measurements in GWs also has disadvantages. For example, usually, values available in GWs are based on the estimations from old measurements and not on the most recent values. Hence, bad assumptions about the data distribution may lead to inaccurate predictions that significantly worsen the quality of information delivered by the GW to the WSN owner.

There is another disadvantage about the quality of the delivered information. Let us suppose, at time t, the GW predicts the measurement that a sensor node will make at time $t + 1$. If the difference between $t + 1$ and t is smaller than the delay to retrieve measurements from sensor nodes, the system will fail to keep the quality of its information when a prediction is not accurate, because it will not have time enough to request the actual measurements made in that instant. For example, a system predicts the measurement that will be made in 1 minute; if its confidence interval does not match the user requirements, it will have to request the real measurement from the sensor nodes; however, if the total time to request and receive the measurement is greater than 1 minute, the GW will fail to attend to the confidence level defined by the user.

6.4. Why (Not) Make Predictions in Sensor Nodes?

As we showed in Sections 3 and 4, it is possible to use the sensor nodes not only to measure information from the external world but also to make predictions and reduce the number of transmissions in a WSN.

6.4.1. Pros. SPSs can avoid unnecessary measurements in sensor nodes that track objects and extend their lifetime, because they are usually equipped with more powerful sensors, such as cameras, microphones, and radio-frequency identification, that require more energy to make a measurement than to process a few machine instructions [Raghunathan et al. 2006].

Regarding the DPSs, the main advantage of exploiting the computing power of the sensor nodes is that since they produce raw data series, the prediction quality can be tested without high communication costs because the sensor nodes are able to check their accuracy locally. Moreover, a recent study has shown that it is possible to successfully substitute real measurements by predictions, reducing the number of transmissions without affecting the quality of the measurements provided by the WSN [Dias et al. 2016].

6.4.2. Cons. SPSs do not fit sensors used to monitor environmental parameters, such as temperature, relative humidity, and solar radiation, because they might spend more energy to predict than to sample the environment. Moreover, since the computing power of these sensor nodes is usually limited, they may not incorporate information about the distant past because of the lack of memory space. As a consequence of the limited computing power and the low amount of information available, complex prediction models may become unfeasible.

Besides that, the absence of updates arriving at the GW may incorrectly imply that predictions are accurate, which requires extra transmissions (e.g., beacons) to ensure their activity status. Therefore, making predictions in sensor nodes requires a reliable transmission scheme in order to make it possible for GWs to provide accurate information.

6.5. Why (Not) Make Predictions in CHs?

As described in Section 2, CHs can be viewed as local GWs placed closer to sensor nodes, because they are responsible for the communication between the sensor nodes and the GW in a WSN. The extra responsibility usually results in higher energy consumption in CHs than in ordinary sensor nodes, provoked by the higher number of transmissions.

If CHs have higher resource availability, it is possible to use more sophisticated prediction methods and exploit the same advantages of making predictions in GWs. In such cases, making predictions in CHs reduces the number of transmissions in WSNs, **improves** the communication in the clusters, and **extends** the sensor nodes' lifetime. In comparison with making predictions in GWs, now the time spent to retrieve measurements from sensor nodes is shorter, thanks to the lower number of hops between CHs and sensor nodes.

Alternatively, some clustering methods are able to periodically elect the CH [Younis et al. 2006]. In such cases, when a sensor node runs out of battery, a decrease in the accuracy of the measurements made by its cluster is expected. Thus, using an accurate prediction model to reduce the number of transmissions may have a larger impact on the quality of the information provided by the WSN. Furthermore, because of the extra responsibilities and the higher number of transmissions, if ordinary sensor nodes act as CHs, their own lifetime will be sharply reduced. Thus, different from the previous case, adopting a data reduction scheme may be necessary to **keep** the WSN alive for a reasonable time.

Note that making predictions in CHs does not preclude predictions in GWs or in sensor nodes. It is possible to make predictions to reduce the transmissions from CHs to GWs and to reduce the transmissions between sensor nodes and CHs, as done in Goel and Imielinski [2001] and Wu et al. [2016].

7. OPEN ISSUES AND FUTURE CHALLENGES

The first research works about prediction-based data reduction in WSNs focused on the most economical ways to process the sensed data and avoid unnecessary transmissions. From these works, two main architectures emerged: (i) in SPSs, either the CHs or the sensor nodes make all predictions, relying on the confidence assessed by the chosen prediction method; and (ii) in DPSs, sensor nodes and CHs make predictions simultaneously and exploit the sensor nodes' proximity to the origin of the data to avoid unnecessary transmissions.

We observed that most of the recent works tended to use DPSs to exploit the best characteristics of each WSN component, that is, the extra computing power of GWs and CHs, and the sensor nodes' proximity to the sources of data. Therefore, the newest contributions focused on finding the most proper prediction methods that (i) fit to limitations imposed by WSNs and (ii) have high precision in several scenarios, such as indoor and outdoor environment monitoring, precision agriculture and structural health monitoring, and so forth. Prediction methods that predict accurately in several scenarios are preferred because they have higher chances of keeping the high accuracy in new deployments.

The work described in Dias et al. [2015] shows the potential of data reduction in an average scenario with a DPS: high-accuracy forecasts can reduce up to 30% the number of transmissions. The described model can be used as a baseline for future works that focus on the data plane to reduce the number of transmissions in sensor networks.

The biggest challenges for the future works involve incorporating characteristics from the statistical theory, considering heterogeneous computing power capabilities, and managing the large-scale use of predictions. These topics are discussed in the following.

7.1. Statistical Theory

In general, current works that use predictions do not refer to any mechanism for data analysis. Hence, statistics considered for data analysis are built before choosing a prediction method and computing prediction models. That is, the environment is supposed to evolve and change in time [Aderohunmu et al. 2013a, 2013b], but this is not considered in many cases, such as in Guestrin et al. [2004] and Li and Wang [2013].

Most of the surveyed works use predictions models in their solutions with very low or absolutely no mathematical basis; that is, authors usually ignore the existence of related works in statistics when deciding which prediction method can be the best one for their scenario, which decreases the reliability of their mechanisms. For example, most of the works [Yann-Ael and Gianluca 2005; Tulone and Madden 2006; Deshpandem et al. 2004; Chu et al. 2006; Guestrin et al. 2004; Debono and Borg 2008; Jiang et al. 2011; Min and Chung 2010; Askari Moghadam and Keshmirpour 2011; Stojkoska et al. 2011; Carvalho et al. 2011; Aderohunmu et al. 2013a; Yin et al. 2015; Raza et al. 2015; Wu et al. 2016] are based on the dataset from the experiments described in Madden [2004]. However, each work makes its own decision about which prediction method to use; that is, none of them incorporate tools to properly analyze the data and find out its characteristics before choosing the prediction method that best fits their requirements. Future approaches may consider the existence of the other prediction methods and the possibility of choosing different methods according to the current state of the environment and its evolution.

Some authors support their assumptions on the statistical theory but do not consider other details inherent to WSNs. For instance, a mechanism to select a prediction model that considers some characteristics of the WSN environments has been presented in Aderohunmu et al. [2013a, 2013b], but the authors did not incorporate the elevated costs to (re)transmit prediction models and their parameters. In conclusion, one challenge for future works is to develop a mechanism to evaluate the efficiency of the use of predictions for data reduction. Different from the current solutions, the new mechanism should be based not only on the communication costs in the evaluation process (as done in Jiang et al. [2011]) but also on (i) the prediction model chosen, (ii) where the predictions will be computed, and (iii) the processing costs implied in their computation.

Adding statistical tools into a mechanism that handles WSNs may be challenging because it involves two distinct areas of knowledge (statistics and computer networks) and a new solution would require a high affinity between their advantages and disadvantages. On the other hand, a mechanism will be significantly more reliable if its actions are based on a statistical theory study of the measured data. Among other benefits, it will be possible to assess its potential improvements in the medium congestion and WSNs' lifetime, as well as lower bounds for the quality of the information that it would produce.

7.2. Heterogeneous Computing Power Capabilities

As we described in Section 2, sensor nodes, CHs, and GWs are expected to have different resources. This may happen not only because of their size but also due to their roles in the system. For example, it is expected that both computational and energy resources of GWs are several orders of magnitude larger than those of the sensor nodes. In other cases, such as in the Internet of Things (IoT), heterogeneous networks may be composed of sensor nodes with distinct memory and computing capacities.

It is an open challenge to exploit the strengths of the different devices in different ways. One possibility is to have different prediction methods running in the same WSN, for example, naive predictions running in sensor nodes with less resource availability, and more complex predictions (e.g., ANNs) in CHs.

Another possibility is to explore the asymmetric characteristic from some prediction methods that use more computationally intensive algorithms to compute models than to make predictions. The ARIMA and the Kernel regression methods are examples of this asymmetry, as shown in Tables I and II. The work of Li et al. [2009] does a similar work based on the ARIMA method using CHs to build and transmit prediction models to sensor nodes, which are responsible for making predictions. This overcomes the current works that take binary decisions and either adopt more complex prediction mechanisms in SPSs or build simpler prediction models when using dual prediction schemes.

7.3. Long-Term Predictions

Existing works mainly focus on predicting measurements that are going to be done in the short term, for example, in the next 5 minutes. The constrained time interval is chosen according to the limitation of the prediction methods (such as AR, MA, and ARIMA), which needs to be often updated in order to produce accurate predictions. In DPSs, inaccurate predictions make WSNs consume much more energy to recompute prediction model parameters and transmit the updates through the network.

Predicting longer time intervals (e.g., 1 hour) may provide a perspective about when the accuracy of the short-term predictions will decrease. That is, based on the extra computational power of the GW, it may be possible to anticipate whether the information produced by the WSN will not meet the minimum quality requirements in the future. As a response, prediction models may be updated, or new prediction schemes can be adopted before a decrease in quality is actually observed.

We expect that long-term predictions may be feasible using external information, which can be either from other WSNs or third-party sources [Oechsner et al. 2014]. This is clearly not a trivial question since decisions made at this level may involve extra transmissions and processing costs to change the sensor nodes' operation. As a tradeoff, it may bring benefits that are not in question in the actual state of the art, in terms of energy savings, medium access, and quality of information, for instance, leading to a new set of mechanisms that focus not only on extending the WSNs' lifetime but also on providing more information to users, which has not been considered as a possibility within the use of predictions so far.

8. CONCLUSION

The study of the state of the art reveals that there is a good reason to invest in making predictions in WSN environments that go beyond adapting the systems' operation to save energy resources and extending the WSNs' lifetime. The future of the IoT depends on the scalability of sensor networks and their capacity to autonomously manage their access to the wireless medium.

With this work, we aim to reduce the gap between the fields of statistics and WSN management and expect that future works in the WSN and IoT fields will be improved by better applying the advantages of the predictions against the limitations of the WSNs. To achieve this goal, we surveyed the existing approaches that use predictions to reduce the number of transmissions in WSNs and explained the prediction techniques that are currently being considered as options in WSNs.

For the complete analysis, we categorized current works according to their architecture, highlighting the challenges to design new mechanisms. For instance, any change in the WSNs' operation must handle two issues: it must detect sensor malfunctioning or changes in the reading dependencies among their measurements, and it must distribute energy consumption among the sensor nodes, which depends on their own predictability and can be unfeasible in some cases. Moreover, we have also shown workarounds for the sensor nodes' hardware limitations, such as extending the sensor nodes' memory capacity and avoiding, in sensor nodes, prediction methods that require complex machine instructions.

Finally, we observed that it is feasible to adopt predictions to reduce the number of transmissions in WSNs. However, it depends not only on the predictions' accuracy but also on the WSN goals, on the sensed phenomena, on the user requirements, and on the architecture adopted to make the predictions.

ACKNOWLEDGMENT

We wish to thank the reviewers for their insightful comments and suggestions that allowed us to improve this article.

REFERENCES

Ameer Ahmed Abbasi and Mohamed Younis. 2007. A survey on clustering algorithms for wireless sensor networks. *Computer Communications* 30, 14–15 (Oct. 2007), 2826–2841. DOI:http://dx.doi.org/10.1016/j.comcom.2007.05.024

Femi A. Aderohunmu, Giacomo Paci, Davide Brunelli, Jeremiah D. Deng, and Luca Benini. 2013a. Prolonging the lifetime of wireless sensor networks using light-weight forecasting algorithms. In *2013 IEEE 8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE, 461–466. DOI:http://dx.doi.org/10.1109/ISSNIP.2013.6529834

Femi A. Aderohunmu, Giacomo Paci, Davide Brunelli, Jeremiah D. Deng, Luca Benini, and Martin Purvis. 2013b. An application-specific forecasting algorithm for extending WSN lifetime. In *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems,*. 374–381. DOI:http://dx.doi.org/10.1109/DCOSS.2013.51

Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19, 6 (1974), 716–723. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1100705

Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdai Cayirci. 2002. Wireless sensor networks: A survey. *Computer Networks* 38, 4 (March 2002), 393–422. DOI:http://dx.doi.org/10.1016/S1389-1286(01)00302-4

Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato, and Hwee-Pink Tan. 2014. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials* 16, 4 (2014), 1996–2018.

Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. 2009. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks* 7, 3 (May 2009), 537–568. DOI:http://dx.doi.org/10.1016/j.adhoc.2008.06.003

Reza Askari Moghadam and Mehrnaz Keshmirpour. 2011. Hybrid ARIMA and neural network model for measurement estimation in energy-efficient wireless sensor networks. In *Informatics Engineering and Information Science*, Azizah Abd Manaf, Shamsul Sahibuddin, Rabiah Ahmad, Salwani Mohd Daud, and Eyas El-Qawasmeh (Eds.). Communications in Computer and Information Science, Vol. 253. Springer, Berlin, 35–48. DOI:http://dx.doi.org/10.1007/978-3-642-25462-8_4

Boris Bellalta, Luciano Bononi, Raffaele Bruno, and Andreas Kassler. 2016. Next generation IEEE 802.11 wireless local area networks: Current status, future directions and open challenges. *Computer Communications* 75 (2016), 1–25. DOI:http://dx.doi.org/10.1016/j.comcom.2015.10.007

Sania Bhatti and Jie Xu. 2009. Survey of target tracking protocols using wireless sensor network. In *2009 5th International Conference on Wireless and Mobile Communications*. 110–115. DOI:http://dx.doi.org/10.1109/ICWMC.2009.25

Alessandro Bogliolo, Valerio Freschi, Emanuele Lattanzi, Amy L. Murphy, and Usman Raza. 2014. Towards a true energetically sustainable WSN: A case study with prediction-based data collection and a wake-up receiver. In *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES'14)*. IEEE, 21–28. DOI:http://dx.doi.org/10.1109/SIES.2014.6871181

George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. 2008. *Time Series Analysis: Forecasting and Control* (4th ed.). Retrieved from http://doc1.lbfl.li/aca/FLMF037168.pdfhttp://onlinelibrary.wiley.com/doi/10.1111/j.1467-9892.2009.00643.x/abstract.

Carlos Giovanni Nunes Carvalho, Danielo G. Gomes, Nazim Agoulmine, and José Neuman de Souza. 2011. Improving prediction accuracy for WSN data reduction by applying multivariate spatio-temporal correlation. *Sensors (Basel, Switzerland)* 11, 11 (Jan. 2011), 10010–10037. DOI:http://dx.doi.org/10.3390/s111110010

Reynold Cheng, Dmitri V. Kalashnikov, and Sunil Prabhakar. 2003. Evaluating probabilistic queries over imprecise data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

David Chu, Amol Deshpande, Joseph M. Hellerstein, and Wei Hong. 2006. Approximate data collection in sensor networks using probabilistic models. In *Proceedings of the 22nd International Conference on Data Engineering*.

Carl J. Debono and Nicholas P. Borg. 2008. The implementation of an adaptive data reduction technique for wireless sensor networks. In *2008 IEEE International Symposium on Signal Processing and Information Technology*. Sarajevo, 402–406.

Antonios Deligiannakis, Yannis Kotidis, Vassilis Stoumpos, and Alex Delis. 2011. Collection trees for event-monitoring queries. *Information Systems* 36, 2 (2011), 386–405. DOI:http://dx.doi.org/10.1016/j.is.2010.08.003

Amol Deshpandem, Carlos Guestrin, Samuel Madden, Joseph M. Hellerstein, and Wei Hong. 2004. Model-driven data acquisition in sensor networks. In *Proceedings of the 30th International Conference on Very Large Data Bases*, Vol. 43. 588–599. DOI:http://dx.doi.org/10.1145/332833.332838

Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner. 2015. The impact of data prediction schemes on the reduction of the number of transmissions in sensor networks. *ArXiv e-prints*.

Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner. 2016. On the importance and feasibility of forecasting data in sensors. *ArXiv e-prints* (April 2016), 30. https://arxiv.org/abs/1604.01275v1.pdf.

David M. Diez, Christopher D. Barr, and Mine Çetinkaya Rundel. 2012. *OpenIntro Statistics*. Retrieved from http://higherlogicdownload.s3.amazonaws.com/AMSTAT/985e96d2-45e9-4950-969e-1bb1b02ac30d/UploadedImages/Newsletters/TSHS_2014_A_Spring_Newsletter.pdf.

Fatih Emekci, S. E. Tuna, Divyakant Agrawal, and A. E. Abbadi. 2004. Binocular: A system monitoring framework. In *Proceedings of the 1st International Workshop on Data Management for Sensor Networks: In Conjunction with VLDB 2004*. 5–9.

Elena Fasolo and Michele Rossi. 2007. In-network aggregation techniques for wireless sensor networks: A survey. *IEEE Wireless Communications* April (2007), 70–87.

Juan A. Garrido-Castellano and Juan J. Murillo-Fuentes. 2015. On the implementation of distributed asynchronous non-linear kernel methods over wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking* 2015, 1 (June 2015), 171. DOI:http://dx.doi.org/10.1186/s13638-015-0382-6

Samir Goel and Tomasz Imielinski. 2001. Prediction-based monitoring in sensor networks: Taking lessons from MPEG. *SIGCOMM Comput. Commun. Rev.* 31, 5 (October 2001), 82–98. DOI:http://dx.doi.org/10.1145/1037107.1037117

Carlos Guestrin, Peter Bodik, Romain Thibaux, Mark Paskin, and Samuel Madden. 2004. Distributed regression: An efficient framework for modeling sensor network data. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, 2004*.

Longjiang Guo, Chunyu Ai, Xiaoming Wang, Zhipeng Cai, and Yingshu Li. 2009. Real time clustering of sensory data in wireless sensor networks. In *Proceedings of the 2009 IEEE 28th International*

Performance Computing and Communications Conference (IPCCC'09). 33–40. DOI:http://dx.doi.org/10.1109/PCCC.2009.5403841

Simon Haykin. 2009. *Neural Networks and Learning Machines* (3rd ed.). Prentice Hall.

Rob J. Hyndman and George Athanasopoulos. 2014. *Forecasting: Principles and Practice*. OTexts.

Rob J. Hyndman and Anne B. Koehler. 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting* 22, 4 (Oct. 2006), 679–688. DOI:http://dx.doi.org/10.1016/j.ijforecast.2006.03.001

C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. 2001. Impact of network density on data aggregation in wireless sensor networks. In *Proceedings 22nd International Conference on Distributed Computing Systems*. 457–458. DOI:http://dx.doi.org/10.1109/ICDCS.2002.1022289

Ankur Jain and E. Y. Chang. 2004. Adaptive sampling for sensor networks. In *Proceedings of the 1st International Workshop on Data Management for Sensor Networks*. 10–16.

Ankur Jain, Edward Y. Chang, and Yuan-fang Wang. 2004. Adaptive stream resource management using Kalman filters. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

Anil K. Jain, Jianchang Mao, and K. M. Mohiuddin. 1996. Artificial neural networks: A tutorial. *IEEE Computer* 29, 31–44. http://www.computer.org/csdl/mags/co/1996/03/r3031.pdf.

Hongbo Jiang, Shudong Jin, and Chonggang Wang. 2011. Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks. *IEEE Transactions on Parallel & Distributed Systems* 22, 6 (2011), 1064–1071.

Ian T. Jolliffe. 2002. *Principal Component Analysis* (2nd ed.). Springer-Verlag, New York. DOI:http://dx.doi.org/10.1007/b98835

Suh Young Kang. 1992. *An Investigation of the Use of Feedforward Neural Networks for Forecasting*. Ph.D. Dissertation. Kent, OH. UMI Order No. GAX92-01899.

Johnsen Kho, Alex Rogers, and Nicholas R. Jennings. 2009. Decentralized control of adaptive sampling in wireless sensor networks. *ACM Transactions on Sensor Networks* 5, 3 (May 2009), 1–35. DOI:http://dx.doi.org/10.1145/1525856.1525857

Emanuele Lattanzi, Matteo Dromedari, Valerio Freschi, and Alessandro Bogliolo. 2013. A sub-a ultrasonic wake-up trigger with addressing capability for wireless sensor nodes. *ISRN Sensor Networks* 2013 (2013), 1–10. DOI:http://dx.doi.org/10.1155/2013/720817

Iosif Lazaridis and Sharad Mehrotra. 2003. Capturing sensor-generated time series with quality guarantees. In *Proceedings of the 19th International Conference on Data Engineering*. 429–440.

Yann-Aël Le Borgne, Silvia Santini, and Gianluca Bontempi. 2007. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Processing* 87, 12 (Dec. 2007), 3010–3020. DOI:http://dx.doi.org/10.1016/j.sigpro.2007.05.015

Wang-chien Lee and Yingqi Xu. 2003. On localized prediction for power efficient object tracking in sensor networks. In *Proceedings of the International Conference on Distributed Computing Systems Workshops*.

Guorui Li and Ying Wang. 2013. Automatic ARIMA modeling-based data aggregation scheme in wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking* 2013, 1 (2013), 85. DOI:http://dx.doi.org/10.1186/1687-1499-2013-85

Ming Li, Deepak Ganesan, and Prashant Shenoy. 2009. Presto: Feedback-driven data management in sensor networks. *IEEE/ACM Transactions on Networking* 17, 4 (2009), 1256–1269.

Chong Liu, Kui Wu, and Min Tsao. 2005. Energy efficient information collection with the ARIMA model in wireless sensor networks. In *Proceedings of the IEEE Global Communications Conference*. 2470–2474.

Hong Luo, Yonghe Liu, and Sajal K. Das. 2007. Routing correlated data in wireless sensor networks: A survey. *IEEE Network* 21, 6 (November-December 2007), 40–47. DOI:10.1109/MNET.2007.4395109

Samuel Madden. 2004. Intel Lab Data. Retrieved from http://db.lcs.mit.edu/labdata/labdata.html.

Azhar Mahmood, Ke Shi, Shaheen Khatoon, and Mi Xiao. 2013. Data mining techniques for wireless sensor networks: A survey. *International Journal of Distributed Sensor Networks* 2013 (2013), Article ID 406316, 24 pages.

Spyros G. Makridakis, Steven C. Wheelwright, and Rob J. Hyndman. 1998. *Forecasting: Methods and Applications* (3rd ed.). Wiley.

Haroon Malik, Ahsan Samad Malik, and Chanchal K. Roy. 2011. A methodology to optimize query in wireless sensor networks using historical data. *Journal of Ambient Intelligence and Humanized Computing* 2, 3 (June 2011), 227–238. DOI:http://dx.doi.org/10.1007/s12652-011-0059-x

Guoqiang Mao, Bar Fidan, and Brian D. O. Anderson. 2007. Wireless sensor network localization techniques. *Computer Networks* 51, 10 (July 2007), 2529–2553. DOI:http://dx.doi.org/10.1016/j.comnet.2006.11.018

Aghileh Djafari Marbini and Lionel E. Sacks. 2004. Adaptive sampling for sensor networks. In *Proceedings of the 1st International Workshop on Data Management for Sensor Networks in Conjunction with VLDB 2004 (DMSN'04)*. ACM Press, New York, New York, 10. DOI:http://dx.doi.org/10.1145/1052199.1052202

Tales Benigno Matos, Angelo Brayner, and Jose Everardo Bessa Maia. 2010. Towards in-network data prediction in wireless sensor networks. In *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC'10)*. ACM Press, New York, NY, 592. DOI:http://dx.doi.org/10.1145/1774088.1774210

David James McCorrie, Elena Gaura, Keith Burnham, Nigel Poole, and Roger Hazelden. 2015. Predictive data reduction in wireless sensor networks using selective filtering for engine monitoring. In *Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications*. Springer New York, New York, NY, 129–148. DOI:http://dx.doi.org/10.1007/978-1-4939-2468-4_6

Jun-Ki Min and Chin-Wan Chung. 2010. EDGES: Efficient data gathering in sensor networks using temporal and spatial correlations. *Journal of Systems and Software* 83, 2 (Feb. 2010), 271–282. DOI:http://dx.doi.org/10.1016/j.jss.2009.08.004

Simon Oechsner, Boris Bellalta, Desislava Dimitrova, and Tobias Hossfeld. 2014. Visions and challenges for sensor network collaboration in the cloud. In *Proceedings of the 2014 8th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. 16–22. DOI:http://dx.doi.org/10.1109/IMIS.2014.3

Mark A. Paskin and Gregory D. Lawrence. 2003. Junction tree algorithms for solving sparse linear systems. *Technical Report* UCB/CSD-03-1271 (2003).

Albert Bel Pereira, José López Vicario, and Gonzalo Seco-Granados. 2012. A pragmatic approach to cooperative positioning in wireless sensor networks. In *Microwave and Millimeter Wave Circuits and Systems*. John Wiley & Sons, Chichester, UK, 135–172. DOI:http://dx.doi.org/10.1002/9781118405864.ch6

Joel B. Predd, Sanjeev R. Kulkarni, and H. Vincent Poor. 2009. A collaborative training algorithm for distributed learning. *IEEE Transactions on Information Theory* 55, 4 (April 2009), 1856–1871. DOI:http://dx.doi.org/10.1109/TIT.2009.2012992

V. Raghunathan, S. Ganeriwal, and M. Srivastava. 2006. Emerging techniques for long lived wireless sensor networks. *IEEE Communications Magazine* 44, 4 (April 2006), 108–114. DOI:http://dx.doi.org/10.1109/MCOM.2006.1632657

Iftach Ragoler, Yossi Matias, and Nimrod Aviram. 2004. Adaptive probing and communication in sensor networks. In *Lecture Notes in Computer Science*. 280–293.

Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.3414

Tifenn Rault, Abdelmadjid Bouabdallah, and Yacine Challal. 2014. Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks* 67 (2014), 104–122. DOI:http://dx.doi.org/10.1016/j.comnet.2014.03.027

Usman Raza, Alessandro Camerra, Amy L. Murphy, Themis Palpanas, and Gian Pietro Picco. 2012. What does model-driven data acquisition really achieve in wireless sensor networks? In *2012 IEEE International Conference on Pervasive Computing and Communications*. IEEE, 85–94. DOI:http://dx.doi.org/10.1109/PerCom.2012.6199853

Usman Raza, Alessandro Camerra, Amy L. Murphy, Themis Palpanas, and Gian Pietro Picco. 2015. Practical data prediction for real-world wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering* 27, 8 (Aug. 2015), 2231–2244. DOI:http://dx.doi.org/10.1109/TKDE.2015.2411594

Yunus Saat. 2011. *Scalable Inference for Structured Gaussian Process Models*. Ph.D. Dissertation. University of Cambridge.

Samer Samarah, Muhannad Al-Hajri, and Azzedine Boukerche. 2011. A predictive energy-efficient technique to support object-tracking sensor networks. *IEEE Transactions on Vehicular Technology* 60, 2 (2011), 656–663.

Silvia Santini and Kay Römer. 2006. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proceedings of the 3rd International Conference on Networked Sensing Systems*. 29–36.

Gideon Schwarz. 1978. Estimating the dimension of a model. *Annals of Statistics.* Retrieved from http://projecteuclid.org/euclid.aos/1176344136.

Yan Shen and Xunbo Li. 2008. Wavelet neural network approach for dynamic power management in wireless sensor networks. In *Proceedings of the 2008 International Conference on Embedded Software and Systems*. 376–381. DOI:http://dx.doi.org/10.1109/ICESS.2008.36

Tossaporn Srisooksai, Kamol Keamarungsi, Poonlap Lamsrichan, and Kiyomichi Araki. 2012. Practical data compression in wireless sensor networks: A survey. *Journal of Network and Computer Applications* 35, 1 (Jan. 2012), 37–59. DOI:http://dx.doi.org/10.1016/j.jnca.2011.03.001

Biljana Risteska Stojkoska, Dimitar Solev, and Danco Davcev. 2011. Data prediction in WSN using variable step size LMS algorithm. In *Proceedings of the 5th International Conference on Sensor Technologies and Applications Fig.*

Sujesha Sudevalayam and Purushottam Kulkarni. 2011. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys and Tutorials* 13, 3 (2011), 443–461. DOI:http://dx.doi.org/10.1109/SURV.2011.060710.00094

Neil H. Timm. 2002. *Applied Multivariate Analysis*. Retrieved from http://link.springer.com/content/pdf/10.1007/b98963.pdf.

Daniela Tulone and Samuel Madden. 2006. PAQ: Time series forecasting for approximate query answering in sensor networks. *Wireless Sensor Networks* 3868 (2006), 21–37.

Mou Wu, Liansheng Tan, and Naixue Xiong. 2016. Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications. *Information Sciences* 329 (2016), 800–818. DOI:http://dx.doi.org/10.1016/j.ins.2015.10.004

Liguang Xie, Yi Shi, Y. Thomas Hou, and Wenjing Lou. 2013. Wireless power transfer and applications to sensor networks. *IEEE Wireless Communications* August (2013), 140–145. DOI:http://dx.doi.org/10.1109/MWC.2013.6590061

Bo Xu and Ouri Wolfson. 2003. Time-series prediction with applications to traffic and moving objects databases. In *Proceedings of the 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access*.

Yingqi Xu, Julian Winter, and W. C. Lee. 2004a. Dual prediction-based reporting for object tracking sensor networks. In *Mobile and Ubiquitous Systems: Networking and Services*.

Yingqi Xu, Julian Winter, and Wang-chien Lee. 2004b. Prediction-based strategies for energy saving in object tracking sensor networks. In *Proceedings of the IEEE International Conference on Mobile Data Management*.

Le Borgne Yann-Ael and Bontempi Gianluca. 2005. Round robin cycle for predictions in wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing Conference* (2005), 253–258.

Yihang Yin, Fengzheng Liu, Xiang Zhou, and Quanzhong Li. 2015. An efficient data compression model based on spatial clustering and principal component analysis in wireless sensor networks. *Sensors (Basel, Switzerland)* 15, 8 (Jan. 2015), 19443–19465. DOI:http://dx.doi.org/10.3390/s150819443

Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian. 2006. Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network* June (2006), 20–25.

Received September 2015; revised May 2016; accepted September 2016