Available online at www.sciencedirect.com

SciVerse ScienceDirect

Energy Procedia 00 (2016) 000–000

Energy Procedia

www.elsevier.com/locate/procedia

8th International Conference on Sustainability in Energy and Buildings, SEB-16, 11-13 September 2016, Turin, ITALY

Predicting large scale fine grain energy consumption

Tania Cerquitelli*

Control and Computer Engineering Department

Politecnico di Torino, C.so Duca degli Abruzzi 24, 10129 Torino, ITALY

Abstract

Today a large volume of energy-related data have been continuously collected. Extracting actionable knowledge from such data is a multi-step process that opens up a variety of interesting and novel research issues across two domains: energy and computer science. The computer science aim is to provide energy scientists with cutting-edge and scalable engines to effectively support them in their daily research activities. This paper presents SPEC, a scalable and distributed predictor of fine grain energy consumption in buildings. SPEC exploits a data stream methodology analysis over a sliding time window to train a prediction model tailored to each building. The building model is then exploited to predict the upcoming energy consumption at a time instant in the near future. SPEC currently integrates the artificial neural networks technique and the random forest regression algorithm. The SPEC methodology exploits the computational advantages of distributed computing frameworks as the current implementation runs on Spark. As a case study, real data of thermal energy consumption collected in a major city have been exploited to preliminarily assess the SPEC accuracy. The initial results are promising and represent a first step towards predicting fine grain energy consumption over a sliding time window.

Peer-review under responsibility of KES International.

Keywords: energy data, big data frameworks, data mining algorithms, data stream analysis

1. Introduction

In the last few years, data generation capability has increased at an unprecedented rate, to such an extent that data rapidly scales towards big data. The abundance of collected information provide an unprecedented opportunity to tackle interesting challenges and add intelligences in real-world applications. The importance of the ability to transform huge data collections into knowledge is indicated by the number of companies and researchers involved in the field of big data. A good example of data producer is represented by energy-related applications which are able to generate large volumes of data that reveal hidden actionable knowledge (e.g., detailed patterns and models to characterize and predict energy consumption) for different interested users (e.g., energy managers, energy analysts, consumers, users living in the building). Since the reduction of wasteful energy consumption is a growing policy

* Corresponding author. Tel.: +039-011-090-7178 ; fax: +039-011-090-7099.
E-mail address: tania.cerquitelli@polito.it

Peer-review under responsibility of KES International.

priority for many countries, innovative systems should be designed to continuously monitor a smart city environment and provide all stakeholders with the tools required to improve energy efficiency.

Being able not only to collect but also to deep analyze large energy-related data volumes could bring to the surface actionable value to support a variety of interested end-users in the decision-making processes. Furthermore, since energy-related applications are good producers of big data collections, analyzing such data opens up a variety of interesting research issues across two research domains: energy and computer science. Extracting actionable knowledge from large volumes of data is a multi-step process that requires considerable interaction between an energy scientist and a computer scientist. Both scientists assume a key role in the analytics process and should be very closely involved in designing innovative and efficient algorithms. The energy scientist needs to define the end-goal, to support the data pre-processing phase, and to assess extracted knowledge. Furthermore, he/she is more strongly involved in the algorithm definition phase which should respect physical laws and correctly model physical events. The computer scientist tackles the task of developing innovative and efficient algorithms, selecting the optimal techniques to achieve the end-goal, looking for a good trade-off between knowledge quality and execution time.

From the computer scientist's point of view, in the data analytics domain, most of the technologies and algorithms involved in big data processing have to be redesigned and adapted to the main features of big data, as well as tailored to the specific features of energy-related data (e.g., heterogeneous data, variable data distribution, data at different abstraction levels, such from fine to coarse grain). In general, applying data mining techniques to big data collections has often entailed coping with a critical bottleneck represented by computational costs. To address this issue, distributed and parallel approaches have been proposed, including such frameworks as Hadoop [1] – the most widely diffused MapReduce implementation – and the Apache Hadoop platform, together with its extensions, such as Apache Spark [2]. MapReduce [3] was designed to cope with very large datasets: its main idea is to break down the processing of the data into independent tasks. Apache Spark [2] is considered the most promising foundation for building an effective data analytics framework. It outperforms Hadoop performance thanks to its distributed memory abstraction. Together with Hadoop and Spark, there are other frameworks supporting the parallelization of data mining algorithms (e.g., GraphLab [4], Google Pregel [5], SimSQL [6]. The exploitation of the above distributed frameworks in the energy domain is challenging because it requires a high level of expertise in computer science and also in the energy domain.

This paper proposes SPEC (Scalable Predictor of Energy Consumption), a data mining-oriented engine aimed at predicting and characterizing energy consumption in buildings. The energy consumption over a sliding time window is analyzed by means of different regression techniques to build a model aimed at predicting the upcoming energy consumption at a time instant in the near future (i.e., a limited time horizon). Each prediction model is tightly tailored to the building efficiency. Both the Artificial Neural Networks (ANN) and the Random Forest Regression (RFR) algorithms have been integrated in SPEC to perform this task of prediction analysis which is computational expensive. The SPEC methodology exploits the computational advantages of distributed computing frameworks as the current implementation runs on Spark to effectively analyze very large data collections.

As a case study, we focus on thermal energy consumption collected in a major city enriched with meteorological conditions. The preliminary performance evaluation demonstrates the effectiveness of the proposed methodology in predicting fine grain energy consumption with a limited average error.

This paper is organized as follows. Section 2 briefly introduces the most used distributed and parallel frameworks. Section 4 proposes the main building blocks of the SPEC engine and thoroughly discusses system exploitation in energy-related applications. Section 5 presents a preliminary evaluation of the SPEC engine on real thermal energy data. Section 3 compares our approach with related works, while Section 6 draws conclusions and presents future work.

2. Distributed frameworks

In recent years, we have been witnessing the diffusion of distributed and parallel approaches, often accompanied with cloud-based services (e.g. Platform-as-a-Service tools) [7] due to the increasing volume of collected data as well as the horizontal scaling in hardware. MapReduce [3] and Apache Spark [2] frameworks provide the high level programming environment allowing programmers to focus only on the algorithmic issues, disregarding low-level details.

MapReduce [3], proposed by Google, has been designed to cope with very large datasets. Thanks to the Hadoop Distributed File System (HDFS), MapReduce takes advantage of data locality, allowing the nodes to process only the data they store. The MapReduce paradigm is designed for batch processing: iterative processes do not fit efficiently since each iteration often requires a new reading phase from the disk. This feature is critical when dealing with huge datasets. Hadoop supports only Java as development language.

Apache Spark [2], instead, has become the favourite platform for large scale data analytics because it overcomes the limitations of the MapReduce paradigm, although it maintains full compatibility with the latter. Spark is a general purpose in-memory distributed platform and it supports Java, Python and Scala as development languages. Differently from MapReduce, Spark enables machines to cache data and intermediate results in memory through the introduction of Resilient Distributed Datasets (RDD), instead of reloading them from the disk at each iteration.

In recent years the success of these distributed platforms has been supported by the diffusion of open source libraries including a wide range of machine learning algorithms. Mahout [8] for Hadoop has been one of the most popular collections of Machine Learning algorithms, containing implementations in the areas such as clustering, classification, and recommendation systems. All the current implementations are based on Hadoop MapReduce. MLlib [9] is the Machine Learning library developed on Spark, and it is rapidly growing. MLlib allows researchers to exploit Spark special features to implement all those applications that can benefit from them, e.g., fast iterative procedures. A variety of machine learning algorithms have already been included in MLlib.

3. Related work

In the last few years the analysis of energy data has received increasing attention from the different and cross research communities, including energy, data mining, databases and statistics communities. The wide exploitation of small and smart sensor devices monitoring indoor and outdoor environmental parameters has been effectively supporting the collection of a large volume of measures with temporal and spatial references. These big data collections have a great potential, because when they are mined an interesting subset of actionable knowledge can be discovered to support the decision making process of facility managers.

A variety of research contributions on these large data volumes of energy related data have been carried out for: (i) identifying the main factors that increase energy consumption (e.g., location [10]); (ii) characterizing consumption profiles among different users [10,11]; (iii) supporting data visualization and warning notification [12]; (iv) efficient storing and retrieval operations based on NoSQL databases [13];

Many research efforts, carried out by computer science researchers, have been devoted to designing and developing systems to provide novel and widespread analytics services based on Big Data technologies. Proposed solutions are general purpose [14] or tailored to a given application domain, such as thermal energy consumption [15], residential energy use [16], renewable energy [17], air pollution levels [18]. The work in [15] discusses the key features of an Energy Management System to support frequent pattern discovering on event streams. A Data Stream Management System (DSMS) is exploited to efficiently support the execution of typical queries of real-time EMSs on time-varying data streams.

Some research efforts have been devoted to characterizing energy consumption at a large scale [19] as well as energy efficiency [20]. The research project described in [19] exploits a NoSQL technology to collect, store and analyze large volumes of energy-related data. In [19], MongoDB [21] has been used as a datawarehouse engine and the MapReduce paradigm has been exploited to compute a variety of basic key performance indicators (KPIs) (e.g., energy consumption per unit of volume) to characterize the energy consumption of single buildings and groups of buildings in the same neighborhood, by considering only the consumption during specific outdoor conditions (temperature range). One step towards the computation of more complex KPIs has been proposed in [20]. The authors in [20] proposed the Energy Signature Analysis (ESA) system which exploits a big data methodology, based on Map-Reduce paradigm, to efficiently compute KPIs to characterize the building's energy efficiency through the energy signature. Two main KPIs were presented: (i) The intra-building KPI to compare latest observations with past energy demand in the same conditions, for example in a similar outdoor temperature and indoor temperature; and (ii) the inter-building KPI to rank the overall building performance with respect to nearby and similarly characterized buildings by considering spatial co-location, building size, and usage patterns (e.g., residential, office, public building). Both KPIs exploit the energy signature of a building that estimates the total heat loss coefficient of a building. The latter is computed by

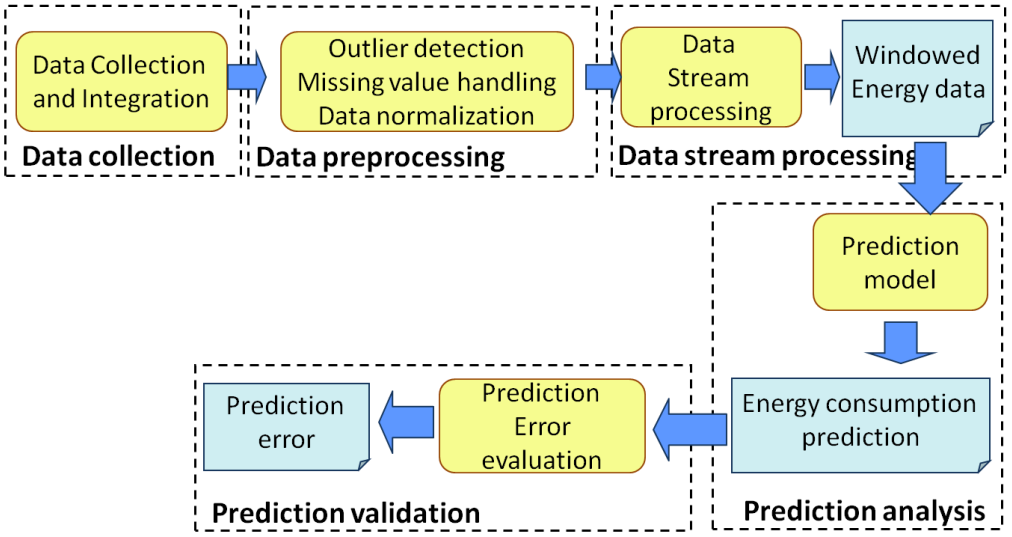

Fig. 1. The SPEC architecture

a linear regression of the power used for heating on the difference between the internal temperature and the external temperature. Instead, the work in [22] presented a centralized engine exploiting exploratory data mining algorithms such as association rules and clustering to characterize energy consumption in buildings. Differently from the above research works [19,20,22], this paper proposes a distributed data mining engine exploiting data mining algorithms to predict fine grain energy consumption. The works cited prevously have a completely different target and analysis approach, and a substantially different architecture (the only similarity lies in the datawarehouse design). Specifically, the target of [19] and [22] is power consumption characterization and the target of [20] is the characterization of energy efficiency to define a building ranking. Whereas this current work aims at predicting energy consumption over a sliding time window. Furthermore, the methodology proposed in [19] and [20] exploits the Map-Reduce paradigm, while this work exploits the Apache Spark implementation.

4. The SPEC engine

SPEC is a distributed engine designed to predict and characterize energy consumption over a sliding time window through data mining. SPEC can be easily exploited in many different energy-related applications including thermal energy consumption, electricity consumption, and other energy-related data. In this paper SPEC is discussed in the context of thermal energy consumption to analyze a variety of energy-related data (e.g., thermal energy consumption, meterological data) as modeled in [19]. Thus, the *Data collection and integration* component collects thermal energy consumption, roughly every 5 minutes, from a large number of smart meters deployed in a smart city. As proposed in [19], energy consumption data are enriched with different meteorological data released as open data through web services. Meteorological data include temperature, relative humidity, precipitation, wind direction, UV index, solar radiation and atmospheric pressure.

To better focus the analysis end-goal (e.g., prediction of fine grain energy consumption in buildings), only a portion of data (e.g., 5-minute energy consumption in a winter season) can be considered. The SPEC components, addressing the main phases of the analysis process, are described in the next sections and shown in Figure 1.

4.1. Data preprocessing

Extracting actionable knowledge from data is a multi-step process, including a preprocessing phase to smooth the effect of possibly unreliable measurements. Preprocessing in SPEC entails three steps: (i) *outlier detection and removal*, (ii) *missing value handling*, and *data normalization* (if necessary).

Outlier detection and removal. An outlier is an observation that lies outside the expected range of values. It may occur either when a measurement does not fit the model under study or when an error in measurement happens (e.g., faulty sensors may provide unacceptable measurements for the thermal energy consumption). To address this issue, SPEC exploits two strategies. (i) *Interquartile range* models the frequency distribution through the computation of median, quartiles, min and max values. The median summarizes the central tendency of the distribution and compared to quartiles provides information about the asymmetry of the distribution. The quartiles give an indication of the variability through the difference interquantile. Values outside the first and third interquartile are not considered in the subsequent analysis steps. (ii) *Leverage* is a coefficient based on the Mahalanobis distance to define whether an energy consumption observation (X_i) is different from the others. For each observation X_i, SPEC computes the leverage as proposed in [23], that is:

$$H_i = Mahalanobis^2(X_i) + \frac{1}{N} \tag{1}$$

where N is the number of energy consumption samples, and the Mahalanobis distance, in our study, is computed as follows:

$$Mahalanobis(X_i) = \sqrt{\frac{(X_i - mean(E))^2}{\sum_j (X_j - mean(E))^2}} \tag{2}$$

where $mean(E)$ is the mean of all energy samples, while $\sum_j (X_j - mean(E))^2$ is the the total square difference between all samples in energy consumption and the corresponding mean.

Only the observations X_i with a leverage value greater than the CutOff threshold are processed in the next analytics step. The CutOff value is calculated as follows:

$$CutOff = \frac{2(K+1)}{N} \tag{3}$$

where K is the number of variables under analysis.

Missing value handling is an important step that significantly affects the mining process. Since we focus on the characterization and the prediction of thermal energy consumption, records with a consumption value equal to zero are disregarded. Instead, to handle missing values on other considered features (e.g., meteorological data), instead, SPEC exploits two strategies: (i) replace them with the daily average value or (ii) replace them with the hourly average value computed in the last week. The choice is mainly driven by the physical meaning of each attribute. For example, case (i) is exploited for the precipitation and wind direction attributes, while case (ii) is for the solar radiation and UV index attributes.

Data normalization When dealing with time series, such as the sequences of thermal energy consumption together with meteorological series, differences in scale and measurement unit exist. Since the normalization technique allows preserving the original data distribution, SPEC integrates two normalization techniques: *min-max* and *z-score*. Since the results of the analysis process can also be affected by the selected normalization step, the end-user can iteratively perform different analysis sessions to identify the strategy that yields better results.

4.2. Data analysis

The core of the data analysis phase in SPEC includes three main blocks: (i) data stream processing, (ii) prediction analysis, and (iii) prediction validation.

4.2.1. Data stream processing

Since thermal energy consumption is monitored roughly every 5 minutes in the district heating system, a large volume of energy data is continuously collected for each building. To efficiently analyze such data the SPEC engine has been designed to perform the energy consumption prediction task for each building separately through the data stream analysis over a sliding time window. Specifically, every time that energy consumption is collected one single sliding time window over the data stream is considered for the prediction task. This window contains a snapshot of the energy consumption monitored in the last instants. It describes the recent past consumption of the building, and

consequently predicts the upcoming energy of the building in the near future. The sliding time window approach requires the definition of the sliding time window size parameter (w_{length}), which determines the temporal context of interest for the analysis. If the time window is very short, then almost instantaneous evaluation of the building's consumption is performed. Instead, a too large time window allows analyzing many data on past building energy performance, but it may introduce noisy information in the prediction analysis. SPEC performs the prediction of energy consumption roughly every 5 minutes. The time window moves jointly with the prediction of the upcoming energy consumption.

4.2.2. Prediction analysis

Different data mining algorithms can be chosen for the prediction analysis of energy consumption. This kind of prediction is a regression task. The aim is to create a model for data under analysis and exploit it for forecasting the upcoming energy consumption at a time instant in the near future. Different methods can be exploited to achieve this, including decision tree, naive Bayes, neural networks, and support vector machines techniques. Each technique employs different learning algorithms to build data models providing a great number of accurate predictions. In SPEC a different data model is created for every time window over the time series considered. To assess the performance of the prediction model, the data under analysis are split into training and test sets. The first is used to build the model, while the second to assess its quality.

Among the available techniques suited to the regression problem (i.e., the prediction of a real value of energy consumption as in this study) we selected *Random forest regression (RFR)* and *Artificial Neural Networks (ANN)*. Both techniques have been widely exploited in many different applications yielding good accuracy performance. The two techniques are briefly presented below by referencing the Apache Spark implementation.

Random forest regression (RFR) is an ensemble learning method easily exploited for regression. Given a training set (i.e., set of records characterized by different input variables with known target values) a multitude of decision trees (data model) is created to support the prediction of the target value. This method significantly reduces the risk of overfitting. In MLlib [9], the RFR algorithm injects randomness into the training process to create a variety of different decision trees minimizing overfitting. The randomness is introduced by (i) performing N subsampling of the original training set to get different training sets (i.e., bootstrapping) to build trees independently, (ii) considering different random subsets of input variables to split at each tree node. Then, the training phase can be performed in parallel. To predict a new record in the test set, a random forest must aggregate the predictions from its set of decision trees assigning a prediction independently. For regression task, each tree predicts a real value. The final predicted value is computed as the average of the independent tree predictions. This strategy reduces the variance of the predictions, improving the performance on test data.

RFR exploits as building blocks the traditional method to create a decision tree and exploit it during the prediction task. We present below the main features of these basic steps. The construction of each decision tree (in a specific data sample) works in a top-down manner, by choosing an attribute test condition at each step that best splits the records. The Gini index impurity-based criterion has been exploited to identify the best way to split the records during the growing of the tree. Each node in the tree specifies a test on an attribute, each branch descending from that node corresponds to a range of possible values for that attribute. Each leaf represents a given value for the target attribute. Starting from the root node of the tree, each node splits the instance space into two or more sub-spaces according to an attribute test condition. Then moving down the tree branch corresponding to the value of the attribute, a new node is created. This process is then repeated for the subtree rooted at the new node, until all records in the training set have been processed. During the prediction phase, each tree model is exploited. Instances in the test set are predicted by navigating the tree from the root down to a leaf, according to the outcome of the tests along the path. The predicted value for a given tree corresponds to the value characterizing the reached leaf of the tree.

Artificial Neural Networks (ANN) simulate biological neural systems. In SPEC we integrated the Scala implementation of ANN allowing the execution and the training of a multi-layer perceptron with two or three levels without restrictions[1]. It includes an input layer, n hidden layers, and an output layer. Each layer is made up of nodes. Each node in a layer takes as input a weighted sum of the outputs of all the nodes in the previous layer, and it applies a nonlinear activation function to the weighted input. The network is trained with backpropagation and learns by itera-

[1] The source code has been downloaded from https://github.com/yannart/Scala-Neural-Network.

tively processing the set of training data records. The network predicts the target value for each training data record. Then weights in the network nodes are modified to minimize the mean squared prediction error. These modifications are made in the backwards direction, that is, from the output layer through each hidden layer down to the first hidden layer.

4.2.3. Prediction validation

This block measures the ability of the SPEC engine to correctly predict the energy consumption values achievable by a building in an upcoming time instant. To this aim SPEC integrates three metrics: (i) *Mean absolute percentage error* (*MAPE*), (ii) *Weighted Absolute Percentage Error* (*WAPE*), and (iii) *Symmetric mean absolute percentage error* (*SMAPE*). The three corresponding expressions are reported below.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - P_i}{A_i}\right| \tag{4}$$

$$WAPE = \frac{\sum_{i=1}^{n}|A_i - P_i|}{\sum_{i=1}^{n} A_i} \tag{5}$$

$$S\,MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - P_i|}{|A_i| + |P_i|} \tag{6}$$

In all formulas, A_i is the actual energy consumption at time t_i while P_i is the corresponding predicted value.

In statistics, MAPE, also known as *mean absolute percentage deviation* (*MAPD*), is exploited to evaluate the quality of a predictor. However, since MAPE represents a percentage error, it is not very suited to model energy prediction error because of a high data distribution variability. Specifically, energy consumption may vary in very long intervals, thus the MAPE error may significantly increase in presence of high energy consumption, and have a limited impact otherwise. The WAPE and SMAPE metrics have been proposed to address this issue. WAPE suffers from not having a specific meaning as an error on the single prediction but only on all the forecasts, while SMAPE is able to correctly model the prediction error for each forecast individually. The only drawback of SMAPE is that it is not symmetric. Thus, overestimated forecasts and underestimated forecasts do not have the same impact. Specifically, for the same value of error prediction the underestimated forecast has a great impact on the overall SMAPE value. Since each metric has benefits and drawbacks, SPEC integrates all of them and leaves the selection to the energy analyst.

5. Preliminary experimental results

We performed a preliminary analysis of energy consumption on a real dataset, including energy consumption of 12 residential buildings, using the SPEC engine. The energy data are related to a complete winter period from October 15^{th} to April 15^{th}. Energy consumption values have been integrated with meteorological information collected from the Weather Underground web service[2] [24]. SPEC creates a predicting model for each building separately.

The datasets have been stored in a cluster at our University running Cloudera Distribution of Apache Hadoop (CDH5.3.1). All experiments have been performed on our cluster, which has 8 worker nodes, and runs Spark 1.2.0, HDFS 2.5.0, and Yarn 2.5.0. The current implementation of SPEC is a project developed in Scala exploiting the Apache Spark framework.

For the prediction task we consider as input variables full time, full date, temperature, humidity, precipitations, pressure, dew point, wind direction, energy consumption in the considered time window to support the prediction of the upcoming *energy consumption in the near future (target)*. For the results reported in this study, the SPEC engine has been configured as follows: Normalization and outlier detection have been performed through the MinMax technique and the Leverage approach respectively. We set the time window size (w_{length}) to 3, and consider only energy consumption in the time frame from 5:00p.m. to 10:00p.m.. To configure the Random forest regressor in MLlib, we

[2] The Weather Underground web service gathers meteorological data from Personal Weather Stations (PWS) registered by users.

set numTrees = 20, featureSubsetStrategy = all, impurity = variance, maxDepth = 4, maxBins = 100. To configure the ANN regressor we set perceptronInputNum = 6 and neuronLayerNum = Array[10, 2, 1].

Preliminary experimental results have been performed to measure the ability of the SPEC engine in correctly predicting the energy consumption values roughly every 5 minutes by analysing the prediction error. Tables 1 and 2 report the MAPE, WAPE, SMAPE values for each monitored building obtained through the artificial neural network (ANN) model and the Random Forest Regression (RFR) model respectively. Since SPEC performs the prediction task roughly every 5 minutes, both predicition models yield good performance. The ANN model trained for each building performs the complex prediction task with a limited error (i.e., MAPE between 6-19%, WAPE 6-10%, SMAPE 3-5% in Table 1). Also the performances of RFR are quite good (i.e., MAPE between 9-20%, WAPE 9-14%, SMAPE 5-7% in Table 2), although slightly worse than the ANN model. These results are promising and demonstrate the potential of the proposed methodology in addressing the cumbersome task of predicting fine grain energy consumption over a sliding window. These results are a first attempt towards the continuous prediction of fine grain energy consumption. There is wide room for improvements: (1) significantly reduce the prediction error and (2) tailor the proposed methodology to predict energy consumption in the transitory time frame (TTF). This TTF is characterized by a large variability of energy consumption spikes which significantly increases the complexity of predicting models.

Table 1. Prediction error for each building: Artificial neural network model.

Building identifier	MAPE	WAPE	SMAPE
B1	0.10412	0.08456	0.03977
B2	0.06391	0.06316	0.03137
B3	0.10046	0.08417	0.04086
B4	0.18848	0.09913	0.04905
B5	0.09055	0.07866	0.03641
B6	0.07931	0.07715	0.03570
B7	0.09087	0.07325	0.03597
B8	0.06407	0.06481	0.03128
B9	0.15671	0.08637	0.04174
B10	0.12616	0.08984	0.04310
B11	0.15229	0.09846	0.04856
B12	0.12401	0.09224	0.04412

Table 2. Prediction error for each building: Random Forest Regression model.

Building identifier	MAPE	WAPE	SMAPE
B1	0.13288	0.11463	0.05578
B2	0.10368	0.10333	0.05129
B3	0.13287	0.11239	0.05576
B4	0.19807	0.11120	0.05797
B5	0.11149	0.10513	0.05194
B6	0.12187	0.12022	0.05903
B7	0.11505	0.09560	0.04743
B8	0.09280	0.09409	0.04621
B9	0.19006	0.11874	0.06082
B10	0.14574	0.11261	0.05553
B11	0.19298	0.13445	0.06539
B12	0.16939	0.13896	0.06958

6. Conclusions

As researchers in data mining algorithms and technologies we believe that statistics, data mining and machine learning techniques provide tools that have a great potential to discover interesting and actionable knowledge. The ability of algorithms to effectively support the decision-making process increases when applied to rich and interesting

data, such as energy-related data. In this paper, we have presented the preliminary version of the SPEC (Scalable Predictor of Energy Consumption) engine to address the fine grain prediction of energy consumption over a sliding time window. Preliminary results, achieved on real data, demonstrate the potential of the proposed approach in generating an interesting and quite accurate prediction model. Currently, we are extending the current version of the architecture towards a *cross-building model* to perform more accurate fine grain predictions. Furthermore, we are working on *adapting the prediction models to a longer time frame*, including the first hours of the morning when a large number of energy consumption spikes occur. Finally, we are *tailoring the* SPEC *engine to other energy-related applications* (e.g., electricity applications).

Acknowledgements

The research leading to these results has partially received funding from the Piedmont Region under the POR FESR 2007/2013 n. 281-79 (EDEN Project).

References

[1] Shvachko K, Kuang H, Radia S, Chansler R. The hadoop distributed file system. In: Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST). MSST'10. Washington, DC, USA: IEEE Computer Society; 2010. p. 1–10.

[2] Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. NSDI'12. Berkeley, CA, USA: USENIX Association; 2012. p. 2–2.

[3] Dean J, Ghemawat S. Mapreduce: Simplified data processing on large clusters. Commun ACM 2008;51(1):107–113.

[4] Low Y, Bickson D, Gonzalez J, Guestrin C, Kyrola A, Hellerstein JM. Distributed graphlab: A framework for machine learning and data mining in the cloud. Proc VLDB Endow 2012;5(8):716–727.

[5] Malewicz G, Austern MH, Bik AJ, Dehnert JC, Horn I, Leiser N, et al. Pregel: A system for large-scale graph processing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. SIGMOD '10. New York, NY, USA: ACM. 2010. p. 135–146.

[6] Cai Z, Vagena Z, Perez LL, Arumugam S, Haas PJ, Jermaine CM. Simulation of database-valued Markov chains using simSQL. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013. 2013. p. 637–648.

[7] Apiletti D, Baralis E, Cerquitelli T, Chiusano S, Grimaudo L. SeaRum: A cloud-based service for association rule mining. In: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013 / 11th IEEE International Symposium on Parallel and Distributed Processing with Applications, ISPA-13 / 12th IEEE International Conference on Ubiquitous Computing and Communications, IUCC-2013, Melbourne, Australia, July 16-18, 2013. 2013. p. 1283–1290.

[8] The Apache Mahout machine learning library. Last Access: June 2016. 2016. URL: http://mahout.apache.org/.

[9] The Apache Spark scalable machine learning library. Last Access: June 2016. 2016. URL: https://spark.apache.org/mllib/.

[10] Depuru S, Wang L, Devabhaktuni V, Nelapati P. A hybrid neural network model and encoding technique for enhanced classification of energy consumption data. In: Power and Energy Society General Meeting, 2011 IEEE. 2011. p. 1–8.

[11] Ardakanian O, Koochakzadeh N, Singh RP, Golab L, Keshav S. Computing electricity consumption profiles from household smart meter data. In: Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference (EDBT/ICDT 2014), Athens, Greece, March 28, 2014. 2014. p. 140–147.

[12] Wijayasekara D, Linda O, Manic M, Rieger CG. Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions. IEEE Trans Industrial Informatics 2014;10(3):1829–1840.

[13] van der Veen J, van der Waaij B, Meijer R. Sensor data storage performance: SQL or noSQL, physical or virtual. In: Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. 2012. p. 431–438.

[14] Zulkernine F, Martin P, Zou Y, Bauer M, Gwadry-Sridhar F, Aboulnaga A. Towards cloud-based analytics-as-a-service (claaas) for big data analytics in the cloud. In: Proceedings of the 2013 IEEE International Congress on Big Data. BIGDATACONGRESS '13. Washington, DC, USA: IEEE Computer Society; 2013. p. 62–69.

[15] Anjos D, Carreira P, Francisco AP. Real-time integration of building energy data. In: 2014 IEEE International Congress on Big Data, Anchorage, AK, USA, June 27 - July 2, 2014. 2014. p. 250–257.

[16] Wang C, de Groot M, Marendy P. A service-oriented system for optimizing residential energy use. In: IEEE International Conference on Web Services, ICWS 2009, Los Angeles, CA, USA, 6-10 July 2009. IEEE; 2009. p. 735–742.

[17] Lu S, Liu Y, Meng D. Towards a collaborative simulation platform for renewable energy systems. In: IEEE Ninth World Congress on Services, SERVICES 2013, Santa Clara, CA, USA, June 28 - July 3, 2013. IEEE Computer Society; 2013. p. 9–12.

[18] Rios LG, Diguez JAI. Big data infrastructure for analyzing data generated by wireless sensor networks. In: Proceedings of the 2014 IEEE International Congress on Big Data. BIGDATACONGRESS '14. Washington, DC, USA: IEEE Computer Society; 2014. p. 816–823.

[19] Acquaviva A, Apiletti D, Attanasio A, Baralis E, Boni Castagnetti F, Cerquitelli T, et al. Enhancing energy awareness through the analysis of thermal energy consumption. In: Proceedings of the Workshops of the EDBT/ICDT 2015 Joint Conference (EDBT/ICDT), Brussels, Belgium, March 27th, 2015. 2015, p. 64–71.

[20] Acquaviva A, Apiletti D, Attanasio A, Baralis E, Bottaccioli L, Boni Castagnetti F., et al. Energy signature analysis: Knowledge at your fingertips. In: 2015 IEEE International Congress on Big Data, New York City, NY, USA, June 27 - July 2, 2015. 2015. p. 543–550.
[21] Chodorow K, Dirolf M. MongoDB: the definitive guide. O'Reilly Media; 2010.
[22] Cerquitelli T, Di Corso E. Characterizing thermal energy consumption through exploratory data mining algorithms. In: Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference, EDBT/ICDT Workshops 2016, Bordeaux, France, March 15, 2016. 2016. p. 1–8.
[23] Blatná D. Outlier in regression. last access: May 2016. 2016. URL: Available at www.laser.uni-erlangen.de.
[24] Weather Underground web service. Last Access: June 2016. 2016. URL: http://www.wunderground.com/.