Predicting energy measurements of service-enabled devices in the future smartgrid

Domnic Savio, Lubomir Karlik, Stamatis Karnouskos
SAP Research, Germany (www.sap.com), Email: domnic.savio@sap.com

***Abstract*—In the future Internet of Things devices will generate massive amounts of data that will flow to enterprise systems and provide a timely view on the execution of business processes. Being able to estimate data generated by devices may have significant effects on planning and execution of business applications. We present some methodologies for mining data gathered from devices in the energy domain i.e. web service enabled smart meters and home appliances. We present here an approach that realise short-term prediction based on neural networks or support vector machines. We consider detailed information about energy consumption coming from service-enabled devices in the broader smart grid envisioned future infrastructure.**

***Keywords*-Internet of Things, short–term prognosis, complex event processing, event prediction, smart grid**

I. Introduction

With mass usage of embedded electronics spawning several domains, things start to get interconnected forming large networks that can monitor and control physical processes. Harnessing the data coming from them, being able to evaluate them in a timely manner and feed the result to business intelligence tools might provide a business advantage. Collecting the information is a major problem today, as no standardized interfaces exist at device level; however we slowly see the applicability of service oriented architecture (SOA) concepts at device level [1]. As a result the mass amount of data that gets generated may be offered to network applications and business services for consumption. This data contains events, status information and commands that needs to be interpreted by data mining technologies to gather information about what is happening on the devices and the processes the participate in.

The ability to predict the behaviour of a device on the basis of its profile and the short history of generated data, might enable us to better plan and fine tune our business services. If prediction models are properly employed, they can enable autonomous systems to make decision based on the dynamic nature of the events. Data mining methodologies and complex event processing in such systems are seen as key components for making decisions in a event driven environment. In this paper we evaluate such methodologies to identify how data mining can enhance business processes by tapping into the power of the Internet of Things.

A. Use Case: Energy consumption

The ever increasing demand on energy and increasing usage of alternative energy sources introduce many challenges that have to be tackled. In the future it is expected that the existing centralized generation of energy will be more distributed, for example the households and factories will not just consume energy, but can also produce it making them Prosumers (Producers and Consumers). The energy demand and production capacity, however is dynamic in nature. Prognostic approaches using the capabilities of the Internet of things in smart houses can enhance electricity trading in markets by estimating energy consumption patterns of a house or factory using historical and relevant data. Today we do not have this capability as only the aggregated values are seen, but with the Internet of Things we can use prediction models that include device specific information.

It is expected that in the smart grid, devices such as the smart meters, home appliances, electric cars etc. will be able to offer their functionality as a service [1]. We have already realized an simulator that can generate a smart city infrastructure based on real-world profiles [1]. The devices generated by the simulator are based on (mobile) software agents and are accessible via web service interfaces. Each device has a control interface and acts autonomously according to its internal goals. As such they represent a complex and highly dynamic event based infrastructure. The devices e.g. smart meters and home appliances like TV, refrigerator and climate conditioners, electric cars etc. consume or produce energy and these measurements are immediately available on a per second basis. The values emitted by each device are based on normalised energy consumption patterns reported by US studies [1].

The ideas that in the future many heterogeneous devices will offer their functionality as a service is not far fetched. Extensive work has been carried out by major industries in R&D projects such as SODA (www.soda-itea.org) and SOCRADES (www.socrades.eu). There the Device Profile for Web Services (DPWS) protocol is used that enables embedded devices to tap to the SOA based infrastructure. Commercially such devices are available off-the shelf for specific functionalities while some more are expected to appear in the near future. The web service interfaces can provide operations to control a device such as turn the device on or off, reduce performance, go to safe mode etc. In this experiment the simulator devices employ the same DPWS stack to host their web services. They provide services that generate energy consumption events typically in frequency of seconds. This data is then used to model the energy consumption patterns by prognosis.

II. Related Work

To estimate the energy consumption in a future time space, based on current information, the pattern of energy consumption has to be identified. The pattern is a function composed of a series of events happening within a period of time[2]. Predicting energy consumption patterns in web service based smart meters can be treated as an estimation problem which is composed of a series of events. Pattern recognition and event prediction methodologies are appropriate solutions [3], [4]. Prediction can be done by time series analysis using various methods such as genetic algorithms or neural networks [5], [6].

An implementation of generic algorithm based application called Timeweaver is noteworthy contribution for prediction of rare events in time-series and event sequences [7]. The event attributes, however, cannot be numerical and genetic algorithms are expensive with respect to time. Hybrid algorithms are used to extract interesting episodes (which is a set of sequences of events in the event stream within a certain interval [8]). MavHome is a project that focuses on building an intelligent environment for living. Based on data from appliances, the prediction models are build so that some daily tasks can be automatized, for example, radio turning automatically on after a toaster is turned on [9]. Common algorithms for event prediction, sequence matching, compression based prediction, prediction using Markov models, and episode discovery are analysed [10]. A neural network is used to combine the results of these algorithms. The model was successfully tested in an home environment with a single inhabitant. Another approach about intelligent environments is using sequential mining based on temporal relations between events [11]. The algorithm is working with simple events, without attributes and was evaluated on small datasets. Efficient algorithms for mining periodic patterns from a sequence of events were studied [12].

The energy consumption estimation has been studied and short-term predictions are compared namely auto-regressive methods, feed-forward neural networks and fuzzy logic approaches. Auto-regressive methods showed much worse results compared to the other two [13]. However better results were obtained with the modified ARIMA method than by using neural networks [14]. Energy prediction using neural networks is a critically reviewed topic [15] as the neural networks are typically over-fitted (over-parametrized), since the input as well as output has typically tens of values. This leads to a huge neural network and the choice of suitable topology is not a trivial task. Furthermore, the evaluation is usually not performed thoroughly. Estimating energy consumption using web services involves data gathering, service monitoring, building prediction models and applying mining algorithms for prognosis. The methodologies in this experiment are fairly generic and can be extended to various other data mining tasks where web services are deployed.

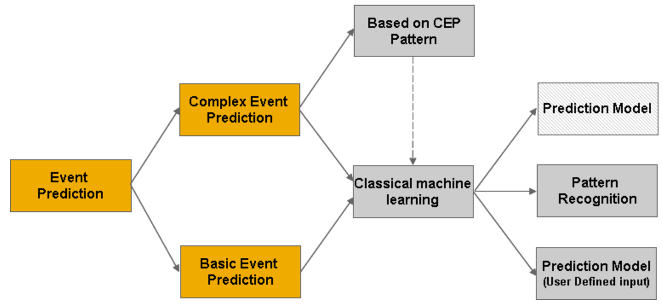

Figure 1. Event Prediction Algorithms

III. Event Prediction

To estimate the energy consumption the respective measurements from the metering service of each device need to be acquired. In our case this is realized on an event based way by having the web service clients subscribing directly to the events from the devices. These WS-Events (according to the WS-Eventing standard of W3C) can be analysed for hidden patterns that reveal the energy consumption information. Events can be related or dynamic, which relies on the user behaviour. There are two types of events in a complex event processing system, basic events and complex events (Figure 1). Basic events are created by sensors, meters, and all other components of the system. A complex event is defined as a pattern that is created from basic events or other complex events. A complex event processing engine uses either rules or queries to define events of interest. When a pattern is recognized in the received event set, an event can be triggered or a set of commands can be executed. The pattern would be then used for complex event prediction. This could be very complex task, because complex event patterns can be expressive in terms of computing time and can contain various logical and temporal operators, functions and filters.

The complex event prediction can be seen as a basic event prediction i.e. ignoring the information included in complex event definition. The naive solution to include all recent event data as an input for the prediction algorithm is not feasible as most of the recent events received by the systems may not influence the predicted result, but would include significant overhead in communication and computation. The input would consist mostly irrelevant data and the space for searching a pattern would be too large. Events relative to the predicted event could be assessed by pattern recognition algorithms. However, patterns that can be discovered by these algorithms, are much simpler compared to complex event patterns. The sequence and time-series pattern recognition algorithms, typically discover simple temporal patters e.g. event A follows event B or event A occurs near event B etc. More over, few algorithms work with numerical event attributes and they operate on small data sets.

Our approach follows a method to built prediction models by considering events selected by the user, who is expected to be a domain expert and knows what type of events may influence the event to be predicted. The relationships and causality between current events and the predicted events will be learned by the prediction model. After a single event is predicted based on historical data, a set of future events or a function of the pattern can be derived with respect to time. For example, the user can select a set of web service devices in his home to estimate energy consumption of these devices in a time space. Alternatively he can select the data from his smart meter device alone, to estimate his total energy consumption for the next 24 hours. This can be extended from simpler prediction requirement to complex scenarios where there are multiple actors for event sources and collaboratively orchestrate an operation. Prediction can be done to estimate the result of the operation in a short future time interval.

A. *Occasional vs. Heartbeat Events*

In the Internet of things, two types of events have been identified i.e. occasional and heartbeat events. Occasional events signify a change of a system state, and do not necessarily occur in periodic intervals. A system state can be inferred from recent events, therefore occasional events may be predicted based on recent events. Heartbeat events are typically sensor data carrying information about environmental characteristics such as temperature or any other continuous value generated usually periodically. Typical usage of the heartbeat events is e.g. to verify the status of the device (i.e. that it is not malfunctioning) or to acquire data at specific intervals. They represent snapshots of a continuous value in a time and have often numerical attributes.

IV. Proposed methods for Prediction

As discussed in section III, prediction relevant events will be selected by the user and the prediction problem can be treated as either a classification (true/false) or value prediction problem. A modular prediction methodology is proposed as shown on the Figure 2. The historical event data is stored in a database and it is assumed that every event type has a dedicated database table and every attributed is mapped to a column including time stamp information. In a real-world application it is expected to have a Complex Event Processing (CEP) engine that transforms data (pattern matching and data transformations) and feed the recent event data to both database and event prediction tool. The CEP engine also ensures that the event prediction tool gets the data for prediction and training from database in the same format.

For each prediction model there can be many single input loaders that provide event data as a parameter for the prediction algorithm. The data from input loaders are then concatenated to a vector. The input is either fed directly to

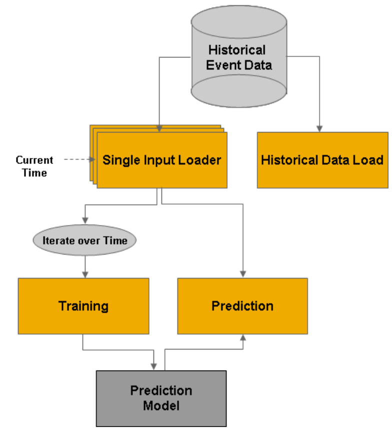

Figure 2. Concept of Prediction Model

prediction model to perform the estimation using prediction model/algorithm (Figure 2) with properly set parameters. The parameters of the model have to be set in training phase, where the historical event data are fed to the prediction model with expected value of prediction for each sample. Once the model is trained, the time is the only parameter; current time is used for future prediction. Therefore the training sample set is generated by iterating over randomly generated times (in history). The input data is prepared for estimation, and expected result is analysed later.

When querying the event data at a certain instance, considering time alone as data is not sufficient. Since the time is continuous, the probability that an event happened exactly at that time is almost zero. Hence an interval is considered that includes a tolerance where an event can be valid. It is time consuming to query for 100 events iterating over time than querying for a single entity. Hence batch scripts to query and load data are used. These scripts were developed in the data mining tool Rapid Miner (http://sourceforge.net/projects/yale). In the event analytical tool it is possible to define prediction models, train them, perform predictions etc. Moreover a prediction function can be defined as a function of prediction models. The function takes as an input the predicted values of the estimation model and aggregates them. In our demonstration scenario the function determines the total energy consumption at a given point of time in the future.

A. *Input Loaders*

The proposed prediction model is flexible. The user can define in rapid miner a customized single input loader. In case of occasional events, the most relevant data for prediction are the recent event data from events that influence the occurrence of predicted event. As an example when event B is predicted and event A influences the occurrence of event B, then last (n) occurrences of event A are most likely relevant. The same holds true for the heartbeat events. A

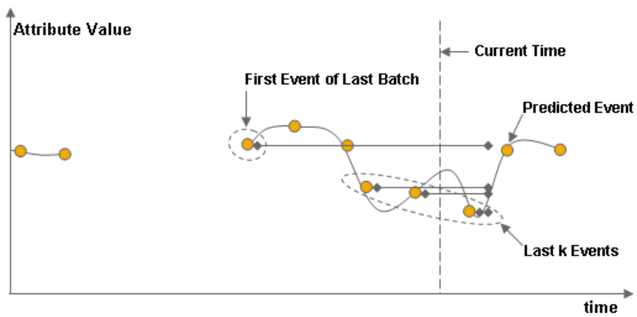

Figure 3. First Of Last Batch Event & Last k Events

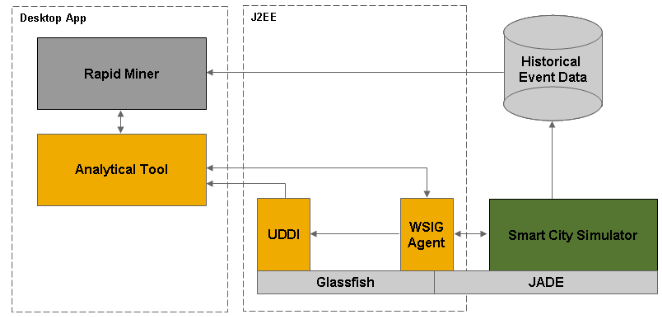

Figure 4. Experimental Architecture

special case is however the time series, where the last (n) values (events) are taken into account when predicting the next value.

For heartbeat events, the called first (event) of last batch (FoLB) can be relevant. Assuming that the device is operating in periodic cycles. Often, especially in the case of home appliances, the length of interval when the device is on, has a certain pattern. To predict when the "ON" part ends, the last n events are not sufficient. The time, how long the device is triggering events is also relevant. This information is considered in the FoLB. Both FoLB and last k events are visualized on the Figure 3.

In both cases, last n event and FoLB, the relative time is important value. If the last event occurred long time ago, the attribute value of this events might be irrelevant. The prediction model is able to recognize this situation based on the relevant time. The scripts to gather last k events and FoLB were designed in rapid miner. In both cases a specified attribute value and relative time are returned.

V. Experimental Setting for Evaluation

As we strive towards a large-scale infrastructure, we use a simulator [1] that creates hundreds of devices that emit events. A smart city is a set of households containing appliances consuming energy, sending events about the consumption in fixed intervals and offering operations to manage the device e.g. turn it on/off.

To set-up the environment as close to the real world scenarios, the system depicted in Figure 4 was developed. The event prediction tool expects operations provided by devices to be offered as web services. The WSIG JADE platform agent is generating web services from all agents that simulate devices on the platform and also transforms WS operation invocations to JADE platform operation invocations. A UDDI server was used as a central register to catalogue the web services hosted by the simulated appliances e.g. metering services. The analytical tool gathers the WSDL files describing web services from the UDDI server and performs a call on the WSIG agent. Dynamic Invocation Interface (DII) is used to construct a web service calls on the fly.

The recent event data for prediction as well as the event data for training are gathered from database. Rapid miner is an open source data mining and machine learning toolbox that is used for data transformation to prepare the input for prediction and training. The machine learning algorithms are used to build prediction models as well as apply them to estimate the prediction. The analytical tool was implemented in Java using the Swing Application Framework to ease the integration and reuse visualization components.

VI. Short-term Energy Consumption Prediction

Traditionally time series analysis is applied along with weather data and time for energy consumption prognosis. Weather and time information are directly fed as an input to a prediction model; different prediction models are built for different weather conditions. We propose a novel approach leveraging detailed information about consumption from each device by dynamically building a prediction model for each device. The model holds the hidden pattern of energy consumption by the device and this pattern can be reused every time a new estimation is required.

Various methods for time series analysis such as neural networks, fuzzy logic, and ARIMA are used. Time series are suitable for a long-term prediction, since they are based on temporal behaviour of overall energy consumption and information about current consumption is irrelevant for the long-term prediction. On the other hand, in short-term prediction, understanding how long the devices are already running and what devices will be turned on or off is very valuable. This information, however, is lost when only the aggregated overall consumption is acquired but not the individual behaviour of each device.

In our approach we leverage detailed information about consumption from each device and build a separate prediction model for every device. We base the aggregated consumption on the sum of distinct consumption predictions of each device. Prediction models for devices are simpler, since the amount of information they have to learn is typically smaller. This can be clearly seen that devices regularly turning on and off such as fridge.

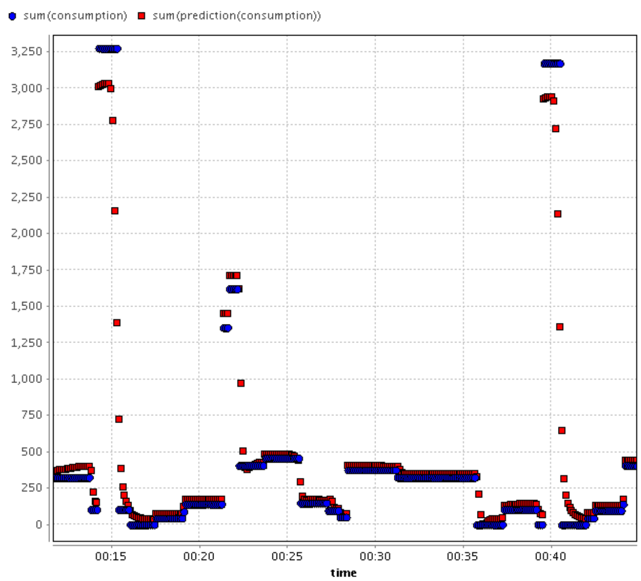

Figure 5. Subset of predicted values of overall consumption compared to real consumption (neural network)

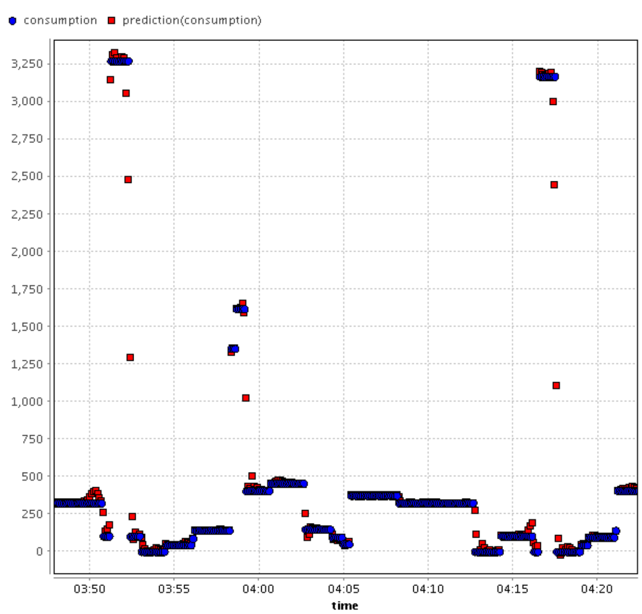

Figure 6. Subset of predicted values of overall consumption compared to real consumption (support vector machine)

To predict consumption for an appliance in smart city, a pair of values have been considered as an input. This includes relative time since the last event from that device, and relative time since first event of last batch. We have configured devices in the smart city simulator to behave periodically; they are turned on for a certain time and turned off of for certain time. The rationale is explained in the section IV-A. Relative time since last event is related to whether the device should be on or off. The relative time from first event of last batch checks whether the device has not been running too long (on average) and most probably will be turned off soon.

	Root Mean Squared Error
Neural Network	184.0 W
Support Vector Machines	55.2 W

Table I
ROOT MEAN SQUARED ERROR OF NEURAL NETWORK AND SUPPORT VECTOR MACHINE

The appliances created by the smart city simulator can provide events every 100 ms. The prediction was done 200 ms in advance. The prediction precision was evaluated on 800 yet unseen samples from 9 appliances of various type. The prediction models were constructed and evaluated using algorithms implementing feed-forward neural networks and support vector machines. The prediction was compared to the actual values (Figure 5). The optimal neural network topology was experimentally obtained. The network has 2 hidden layers with 20 and 15 neurons and it was trained in 15000 cycles using the back-propagation algorithm. The learning rate was set to 0.7 and momentum to 0.1. The cost parameter C had to be set for support vector machines. Value 2000 was experimentally set as sufficient to obtain precise predictions.

The prediction seems to be fairly precise, however, there is a systematic error in prediction that is usually higher than the real value when the consumption is low, and the error is lower when the consumption is high. More over, prediction models for devices that consume much energy (approx. 3 kW) can predict when the device will be turned on, but the predicted value is much smaller that the real one (Figure 6) and overall consumption prediction is not that precise. We consider that this happens because the values are normalized before they are used as an input for the neural network. Normalization in large intervals may lead to rounding errors or small differences between input values.

Such energy estimation can be be of interest to large data centres who could estimate energy consumption using smart devices in their server blades and in the building infrastructure. Estimated energy information can be used to trade energy in advance on future electricity markets. The methodologies used in this scenario can be easily extended and adapted for similar use cases for example to estimate the production efficiency on a shop floor [16].

VII. RESULTS OBTAINED

The prediction is reasonable and is achieved using neural network and support vector machines. Support vector machines perform better compared to methodologies using neural networks (Table I).

The table II shows time taken while preparing training data, training itself and the prognosis. The prediction was

	Neural Network	SVM
Training Sample Preparation	13:51	13:51
Training	00:38	01:15
Prediction (800 times)	11:07	10:54

Table II
TIME SPENT FOR TRAINING DATA PREPARATION, TRAINING, AND PREDICTION (HOURS:MINUTES).

done on 9 devices. For each device a training sample set with 1000 samples was created. After a training period, the prediction was done 800 times. The time for preparing sample data set and prediction is very high. Rapid miner scripts are used for retrieving data. Some rapid miner operators are inefficient and costly. In addition every time a script is called a databased query is performed.

VIII. CONCLUSION AND FUTURE RESEARCH

The event prediction tool was designed and implemented to handle common event prediction tasks in complex event systems. This data mining approach was evaluated in a network of devices hosting web services that provide access to their internal measurements (i.e. energy consumption) via an event-based way. A smart city simulator was used to create several devices and evaluate the tool. Per-device approach for energy consumption prediction was proposed and evaluated. The results show that the prediction can be done fairly precisely using neural networks and support vector machines. Large attribute domains decrease precision of the prediction models. The prediction is biased when the overall consumption is either too low or too high.

The tool retrieves the data for prediction from database. Using a CEP engine the prediction would have been much faster. Therefore as future direction we see the enhancement of the concept to provide more prediction models for a wider range of appliances hosting various services, rather than single one, and to be able to predict several future values and estimate the consumption curve.

REFERENCES

[1] S. Karnouskos and T. N. de Holanda, "Simulation of a smart grid city with software agents," in *European Modelling Symposium (EMS 2009), Athens, Greece*, 25-27 Nov. 2009.

[2] C. L. Giles, S. Lawrence, A. C.   Tsoi, and C. L. Giles, "Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference," *MACHINE LEARNING*, vol. 44, pp. 161–183, 2001.

[3] J. F. Roddick and M. Spiliopoulou, "A Survey of Temporal Knowledge Discovery Paradigms and Methods," *IEEE Trans. on Knowl. and Data Eng.*, vol. 14, no. 4, pp. 750–767, 2002.

[4] F. Mörchen, "Unsupervised pattern mining from symbolic temporal data," *SIGKDD Explor. Newsl.*, vol. 9, no. 1, pp. 41–55, 2007.

[5] Richard J. Povinelli, Xin Feng, Senior Member, and Senior Member, "A New Temporal Pattern Identification Method for Characterization and Prediction of Complex Time Series Events," in *In IEEE Transactions on Knowledge and Data Engineering*. IEEE Press, 2003, pp. 339–352.

[6] R. J. Frank, N. Davey, and S. P. Hunt, "Time Series Prediction and Neural Networks," *J. Intell. Robotics Syst.*, vol. 31, no. 1-3, pp. 91–103, 2001.

[7] Gary M. Weiss, "Timeweaver: a Genetic Algorithm for Identifying Predictive Patterns in Sequences of Events," in *In Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 1999, pp. 718–725.

[8] A. Srinivasan, D. Bhatia, and S. Chakravarthy, "Discovery of interesting episodes in sequence data," in *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*. ACM, 2006, pp. 598–602.

[9] Diane J. Cook, Michael Youngblood, Edwin O. Heierman III, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja, "MavHome: An Agent-Based Smart Home," *Pervasive Computing and Communications, IEEE International Conference on*, vol. 0, p. 521, 2003.

[10] D. J. Cook, "Artificial intelligence on the body, in the home, and beyond," in *BodyNets '08: Proceedings of the ICST 3rd international conference on Body area networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–6.

[11] Vikramaditya R. Jakkula and Diane J. Cook, "Using Temporal Relations in Smart Environment Data for Activity Prediction," *24th International Conference on Machine Learning*, 2007.

[12] Kuo-Yu Huang and Chia-Hui Chang, "Mining Periodic Patterns in Sequence Data," in *Data Warehousing and Knowledge Discovery, 6th International Conference, DaWaK 2004, Zaragoza, Spain, September 1-3, 2004, Proceedings*, ser. Lecture Notes in Computer Science, Yahiko Kambayashi and Mukesh K. Mohania and Wolfram Wöß, Eds. Springer, 2004, vol. 3181, pp. 401–410.

[13] K. Liu, S. Subbarayan, R. R. Shoults, M. T. Manry, C. Kwan, F. I. Lewis, and J. Naccarino, "Comparison of very short-term load forecasting techniques," *Power Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 877–882, 1996.

[14] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," *Power Systems, IEEE Transactions on*, vol. 16, no. 3, pp. 498–505, 2001.

[15] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 44–55, 2001.

[16] D. Yazar and A. Dunkels, "Efficient Application Integration in IP-based Sensor Networks," in *Proceedings of ACM BuildSys 2009, the First ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, Berkeley, CA, USA, Nov. 2009. [Online]. Available: http://www.sics.se/~adam/yazar09efficient.pdf