Applied Energy 193 (2017) 369–380

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Multiscale stochastic prediction of electricity demand in smart grids using Bayesian networks

Nastaran Bassamzadeh [a,*], Roger Ghanem [b]

[a] Department of Civil Engineering, University of Southern California, Los Angeles, CA 90089, USA
[b] Viterbi School of Engineering, University of Southern California, Los Angeles, CA 90089, USA

HIGHLIGHTS

- A probabilistic load forecasting model using Bayesian networks is proposed.
- Model learns dependencies between variables without making prior assumptions.
- The impact of real time pricing on consumption behavior of customers is studied.
- We investigate model performance at varying spatio-temporal levels of aggregation.

ARTICLE INFO

Article history:
Received 29 August 2016
Received in revised form 3 January 2017
Accepted 11 January 2017

Keywords:
Bayesian networks
Forecasting
Machine learning
Prediction
Probabilistic modeling
Smart grid

ABSTRACT

Demand management in residential buildings is a key component toward sustainability and efficiency in urban environments. The recent advancements in sensor based technologies hold the promise of novel energy consumption models that can better characterize the underlying patterns.

In this paper, we propose a probabilistic data-driven predictive model for consumption forecasting in residential buildings. The model is based on Bayesian network (BN) framework which is able to discover dependency relations between contributing variables. Thus, we can relax the assumptions that are often made in traditional forecasting models. Moreover, we are able to efficiently capture the uncertainties in input variables and quantify their effect on the system output. We test our proposed approach to the data provided by Pacific Northwest National Lab (PNNL) which has been collected through a pilot Smart Grid project.

We examine the performance of our model in a multiscale setting by considering various temporal (i.e., 15 min, hourly intervals) and spatial (i.e., all households in a region, each household) resolutions for analyzing data. Demand forecasting at the individual households' levels is a first step toward designing personalized and targeted policies for each customer. While this is a widely studied topic in digital marketing, few researches have been done in the energy sector. The results indicate that Bayesian networks can be efficiently used for probabilistic energy modeling in residential buildings by discovering the dependencies between variables.

1. Introduction

Smart Grid which represents the future generation of power systems, is composed of multiple interacting systems such as renewable supply network, distribution and storage systems and communication systems. The underlying idea which has entailed this transition is the growing need for optimized energy consumption and management. This goal is sought with the aid of digitized systems and sensors which are capable to monitor the system and collect relevant data at various scales. The information network, ensures the viability of data-driven and data-aware perspectives, both on the side of utilities and consumers. This enables both consumers and utilities to share in the responsibility and benefits of access to advanced technology. Being constituted of multiple stochastic, dynamic and distributed components, the need for an automated and intelligent framework to predict the overall system behavior in Smart Grids is accentuated. In this paper, we aim to enhance our stochastic modeling capability in complex systems by adapting a data-driven approach.

* Corresponding author.
E-mail addresses: Bassamza@usc.edu (N. Bassamzadeh), Ghanem@usc.edu (R. Ghanem).

http://dx.doi.org/10.1016/j.apenergy.2017.01.017
0306-2619/

Predictive modeling such as load forecasting is one of the core problems that is studied in power grids. However, the advent of smart meters in the context of Smart Grids introduces some new aspects to this traditional problem. Smart meters facilitate a real time interaction between power supplier and households in terms of collecting high frequency consumption data from households as well as sending incentive prices to the customers. The utility companies seek to manage demand at the customers' side by communicating incentives to them such as real time electricity prices. The pricing policy is aimed at shifting the demand from peak hours to off-peak hours. Accordingly, we encounter a huge amount of high resolution data collected at various time scales at the households' level. The rich entity of data enables decision makers to design personalized policies for each specific household. Thus, the availability of these high resolution data introduces potential benefits as well as some complexities to predictive problems in Smart Grid.

The particular characteristics of Smart Grids call for a new predictive framework that is able to efficiently capture these features. First, we need a predictive model that well suits high dimensional settings where the number of predictors is large. Historical consumption, weather variables, time parameters and real time price signals are among the contributing variables in predicting demand. Moreover, the availability of data at varying spatial and temporal granularities introduces new challenges in terms of multiscale models and the effect of model resolution on the results. Furthermore, given the detailed customers' consumption data, utility companies are interested in having models at the customers' scale which are able to recommend personalized policies. Finally, quantifying the uncertainty associated with the involved components and formalizing their effect on the prediction result in not a trivial task in this context.

In this paper, we propose Bayesian network framework as a probabilistic tool to model the dependencies between various contributing factors in demand forecasting in the Smart Grid context. Bayesian networks provide a single framework for density estimation, probability propagation and inference in complex systems. It is argued that Bayesian networks are an appropriate tool for probabilistic modeling in complex systems when the number of variables is high. In this regard, we seek to probabilistically predict demand subject to real time prices (as well as other factors) at multiple spatial and temporal scales. The load forecasting problem is studied at 15 min and hourly time resolutions. Here, the goal is to find the optimal time interval for sending, receiving and analyzing the data to/from customers. Besides, we develop a model to predict the demand at the households' scale which can be used as an aid for personalized policy making. We demonstrate the performance of our model using a real data set provided by Pacific Northwest National Lab (PNNL).

Therefore, the main contribution of this paper is to investigate whether Bayesian network model can be applied to sensor based probabilistic load forecasting problem. This modeling framework has significant implications. First, the structure of the model is purely learned from the data and no prior assumptions are inserted. Second, the high number of influencing variables as expected in sensor based models will not pose any modeling challenges, since BNs well adopt to high dimensional settings. Moreover, introducing new variables to the model can be well digested by the algorithm. Third, the full probability distribution of the demand variable can be inferred even if all predictor variables are not observed. Fourth, prediction at high granularity levels, specifically at the households' scale can be conducted which has major benefits for utility companies in designing targeted and personalized policies.

Another novelty of our work owes to the data set that we use. The scarcity of high resolution sensor data at household level makes any such data set worthy. Our data set is consisted of consumption values at 5 min intervals for 25 customers over a year. Also, the real time price that has been communicated to customers is available at 15 min intervals. This provides a unique opportunity to investigate whether pricing policy has been effective in reducing or changing the consumption behavior of customers.

2. Related work

Load forecasting problem has been extensively addressed in the literature. Generally speaking there have been two main lines of research. The first direction of research is based on "parametric" statistical models such as time-series and linear regression (e.g., [1,2],3). The second trend encompasses "nonparametric" models such as support vector regression, neural networks and fuzzy logic models [4,5]. A comprehensive literature survey of load forecasting techniques is presented in [6]. Despite being intuitive, parametric methods often suffer from the limitations posed by the underlying assumptions in their construction. The form of the parametric model is set in advance and thus it can't incorporate unspecified relationships between the target variable and the predictors. Moreover, these models are not generally suited for high dimensional problems (when the number of predictors grow) since they will need a prohibitively large number of parameters to be calibrated. Additionally, these models fail in treating the fine resolution data provided by the smart sensors at households' levels.

Nonparametric models which are heavily based on machine learning algorithms, have gained much popularity for sensor based energy forecasting [7]. In this data-driven approach, data from various sources such as energy smart meters, weather stations and resident occupancies are fed into a machine learning algorithm aiming to find the model that gives the best match between the model output and the observed data. In this category, the two most common algorithms used for sensor based energy forecasting are *artificial neural networks* [8,9] and *support vector machines* [10,11]. A comprehensive study of the performance of various machine learning algorithms for load forecasting is presented in [7].

Scarcity of sensor based data for residential buildings when contrasted to commercial buildings has caused the majority of sensor based energy forecasting literature to be focused on commercial buildings [12]. Moreover, most sensor based models for residential buildings are conducted based on monthly data gathered from monthly utility statements. Therefore, there is a need to explore additional techniques for modeling residential energy consumption with higher granularity data sets and provide more insight regarding appropriate modeling approaches for this problem. Residential load forecasting with higher granularity data sets is studied in [13,14].

While deterministic load forecasting is quite valuable, possible sensor errors, intrinsic weather uncertainties and renewable integration requirements mean that probabilistic load forecasting needs to be adopted more in energy planning and operations. The literature on probabilistic load forecasting (PLF) is quite limited especially when comoared to probabilistic wind power forecasting (PWPF) [15]. However, [16] claims that PLF should be regarded just as important as PWPF in the utility industry. The authors argue that typical 15% error in day ahead wind power forecasting, where wind penetration is around 20%, gives a similar absolute error as 3% error in day ahead load forecasting for a medium sized US utility with an annual peak of $1GW$-$10GW$. It is claimed that reducing the forecasting error by only 1%, in terms of mean absolute percentage error (MAPE), can save hundred thousand dollars per GW peak for a utility company [16]. Thus, it is quite evident that uncertainty quantification and probabilistic

modeling in load forecasting is a much needed and missing puzzle piece that needs to be thoroughly addressed.

The literature on probabilistic load forecasting is not well developed. The authors in [17] propose an approach based on Gaussian process with hierarchical Bayesian estimation for probabilistic one-step-ahead daily peak load forecasting. Moreover, a modified bootstrap method for simulating the forecasting residuals and then generating prediction intervals for the electricity demand has been studied in [18]. Probabilistic load forecasting in smart grids has been addressed in [19] using a Bayesian approach. Vlachopoulou et al. [20] use dynamic Bayesian network to predict water heater load variations over time. Bayesian network framework has also been used in citebnpredict5 for predictive control of smart buildings.

The remainder of this article is organized as follows: in Section 3, we introduce the concept of Bayesian networks and provide necessary notations to learn and use the model as a probabilistic predictive model. Section 4 discusses the problem and develops Bayesian network model to forecast demand at multiple spatial and temporal resolutions. We conclude the paper in Section 5.

3. Bayesian networks

A Bayesian network (BN), is a directed acyclic (i.e. having no cycles) graphical (DAG) model that encodes the joint probability distribution of a set of random variables by making conditional independence assumptions [21]. Bayesian networks have become popular representations for extracting knowledge from data in complex uncertain systems in recent years. According to [22], BNs have at least four remarkable advantages over the state-of-the-art data analysis techniques for density estimation, regression, classifications and clustering. First, since BNs are able to encode the dependencies between the input variables, they can easily handle incomplete datasets. Second, in a BN model the causal relationships between the variables can be learned. This provides valuable insights regarding the specific domain under study. Third, Bayesian networks provide an intuitive way to incorporate the prior expert knowledge into the model. This is achieved by using Bayesian statistical techniques in modeling. Finally, BNs provide a principled framework to avoid overfitting of data. A typical BN is shown in Fig. 1.

In this paper, we consider a Bayesian network over a finite set of discrete random variables, $X = \{X_1, \ldots, X_n\}$. A BN represents a joint pdf over X by encoding conditional independence assertions as well as a collection of pdfs. The assertions of conditional independence are modeled through a directed acyclic graph (DAG) structure. Thus, we use the pair $B = \langle S, \Theta \rangle$ to define a BN where S is the BN structure (a DAG) in which nodes represent random variables $X_1, \ldots, X_n$ and edges express direct dependencies between the variables. Moreover, Θ is a set of pdfs corresponding to that structure.

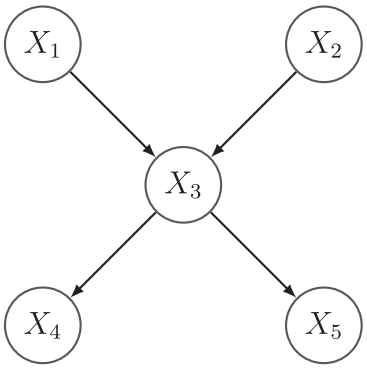

Fig. 1. An example BN [23].

The chain rule of probability implies that a joint distribution can always be represented as follows, using any ordering of the variables:

$$\rho(X_1, \ldots, X_n) = \rho(X_1)\rho(X_2 \mid X_1)\rho(X_3 \mid X_2, X_1) \ldots \rho(X_n \mid X_{1:n-1}) \quad (1)$$

As n gets large, evaluation of $\rho(X_i \mid X_{1:i-1})$ becomes more and more complicated. In a multinomial setting where each variable has K states, $\rho(X_1)$ is represented as a table of $O(K)$ numbers (In fact, there are only $K-1$ free parameters, but we use $O(K)$ for simplicity). Similarly $\rho(X_2 \mid X_1)$ is a table with $O(K^2)$ parameters. Thus, there will be $O(K^n)$ parameters in the model, learning which requires lot of data.

The key solution to characterizing large joint distributions is to make some assumptions about conditional independence (CI). We say that X and Y are conditionally independent given Z, denoted by $X \perp Y \mid Z$ if and only if:

$$X \perp Y \mid Z \iff \rho(X, Y \mid Z) = \rho(X \mid Y)\rho(Y \mid Z) \quad (2)$$

Assume that the structure of the directed acyclic graph is known. There exists efficient algorithms to find the ordering of the nodes such that parents come before children [24]. The parent and children notion in a directed graph coincides with their common sense meaning, i.e. parent of a node is the set of all nodes that feed into it and the children of a node is the set of all nodes that feed out of it. Given this ordering, which is known as topological ordering, **ordered Markov Property** is defined as the assumption that a node only depends on its immediate parents, not all predecessors in the ordering [25], so:

$$X_s \perp X_{pred(s) \setminus pa(s)} \mid X_{pa(s)} \quad (3)$$

where $pa(s)$ are the parents of node s, and $pred(s)$ are the predecessors of node s in the ordering. This is in fact a generalization of the first order Markov assumption where we assume that variable at time $t+1$ is independent of all variables up to time $t-1$ given that variable at time t is observed. This assumptions generalizes Markov property from chains to DAGs. Accordingly, the chain rule (1) for the topological ordering is reduced to a factorized expression given by:

$$\rho(X_1, \ldots, X_n) = \prod_{i=1}^{n} \rho(X_i \mid pa(X_i)) \quad (4)$$

where each term $\rho(X_i \mid pa(X_i))$ is a conditional probability distribution (CPD). Here, we have used the relation $\rho(X_i \mid X_1, \ldots, X_{i-1}) = \rho(X_i \mid Pa(X_i))$ based on ordered Markov assumption to reduce the model. This factorized joint distribution only holds if the CI assumptions encoded in the associated DAG are correct. In case each node has $O(F)$ parents and K states, there will be $O(nK^F)$ parameters in the model which is much less than the $O(K^n)$ needed by a model which makes no CI assumptions.

In the discrete setting, we associate with each node in the graph a conditional probability table (CPT) which tabulates $\rho(X_i \mid Pa(X_i))$. By convention, each row in the table corresponds to a specific configuration of the parent variables and has a separate multinomial distribution. Assume variable X_i and its parents Pa_{X_i} have r_i and q_i mutually exclusive configurations respectively. Thus, the table has $r_i \times q_i$ entries covering all possible combinations [26]. Each entry in the table is denoted by $\theta_{ijk} = p(X_i = k \mid Pa(X_i) = j)$ which is the probability that node X_i is in state k given that its parents are in state j. Similarly the set of parameters for each row is denoted by Θ_{ij} and the set of all the parameters of the BN model is represented by Θ.

3.1. Learning Bayesian networks

Learning a Bayesian network consists of learning the graph structure and the corresponding CPT parameters that best fit the available data. In general, Bayesian network structure learning has two main applications: knowledge discovery and density estimation. By knowledge discovery we mean that a BN structure can efficiently reveal the conditional independencies between variables. This topic will be elaborated in Section 3.3. On the other hand, the BN structure can be used in conjunction with the CPT parameters to uniquely specify a joint density function over the random variables. These two tasks can not be carried out independently. First, in order to learn the structure, we need some estimation of the parameters to quantify the goodness of fit. On the other hand, to estimate the conditional probabilities, we must know the graphical structure in advance.

Generally, there are two categories of BN structure learning. The first one is search-and-score which assigns a score to each possible BN structure based on its match to the observed data and then searches for the best network. The other one is called constraint-based algorithm, which runs conditional independence tests to learn the structure. In this paper, we are only interested in the search-and-score approach. In this setting, learning a Bayesian network consists of a search conducted over some search space to optimize a scoring metric. The scoring functions are based on different principles, such as information theory and minimum description length [27,28], or Bayesian methods [29,30]. As far as the search algorithm is concerned, since finding the globally optimal graph takes exponential time in most cases, some heuristic search method is usually used. This can be greedy search (such as hill-climbing or tabu search), simulated annealing or genetic algorithms to name the most popular ones. At last, the search space is dominantly the DAG space, however, it can be either equivalence classes of DAGs or orderings over the network variables (an equivalence class refers to a set of DAGs in which all the members imply the same set of conditional independence assertions).

In this paper, we focus on the Bayesian score for discrete Bayesian networks and use tabu search to find the best DAG structure. Let $D = \{C_1, \ldots, C_m\}$ be the set of m observed instances of C (where each case C_i assigns values to one or more variables in X). Our goal is finding a network S that *best matches* the dataset D. The derivations in this section are mostly based on [30]. Posterior probability of a network structure may be used as a Bayesian measure to score a structure:

$$\rho(S \mid D) = \frac{\rho(D \mid S)\rho(S)}{\sum_S \rho(D \mid S)\rho(S)} \tag{5}$$

in which the denominator is a normalization constant. Calculating this constant requires summing over all possible structures which is computationally expensive even for small domains. Thus, we resort to computing the numerator as the network score. Thus, the Bayesian score is defined as $\rho(D,S) = \rho(D \mid S)\rho(S)$. We start deriving the Bayesian score by calculating $\rho(D \mid S)$. Applying the chain rule, we obtain

$$\begin{aligned}\rho(D \mid S) &= \prod_{l=1}^{m} \rho(C_l \mid C_1, \ldots, C_{l-1}, S) \\ &= \prod_{l=1}^{m} \int_{\theta} \rho(C_l \mid \Theta, S)\rho(\Theta \mid C_1, \ldots, C_{l-1}, S)\end{aligned} \tag{6}$$

Here, Θ is dependent on structure S, but we skip this notation for brevity. Thus, to calculate $\rho(D \mid S)$ we sum over all possible values of model parameters. The first term in the above integral is the probability of observing a particular case given that the model is fully specified (known structure and known parameters). Here, we assume that each observed case is a multinomial sample from some BN with some structure and parameters. Moreover, if we assume that the dataset is complete (no missing or hidden variables), then using the factorized expression for joint distribution in BNs as in (4), we have

$$\rho(C_l \mid \Theta, S) = \prod_i \prod_j \prod_k \theta_{ijk}^{\alpha_{lijk}} \tag{7}$$

where α_{lijk} is 1 if and only if $X_i = k$ and $Pa(X_i) = j$ in case C_l and 0 otherwise. To simplify the second term in the integral, researchers assume **parameter independence** which implies that the parameters associated with each node in the BN (each variable) can be computed independent of the other nodes. This is called global independence. Moreover, in each node the parameters of each row of the CPT table (associated with each particular configuration of the node parents) can be determined independent of other rows. This is referred to as local independence. These assumptions lead to the following

$$\rho(\Theta \mid S) = \prod_i \prod_j \rho(\Theta_{ij} \mid S) \tag{8}$$

where Θ_{ij} is the set of parameters of node i when its parents are in configuration j. Plugging (7) and (8) into the integral in (6) and doing some manipulations yields [30]

$$\rho(D \mid S) = \prod_i \prod_j \prod_k \prod_l E(\theta_{ijk} \mid C_1, \ldots, C_{l-1}, S)^{\alpha_{lijk}} \tag{9}$$

where E denotes the expectation of θ_{ijk} with respect to $\rho(\Theta_{ij})$. To calculate the expectation in (9) we need to know $\rho(\Theta_{ij} \mid D, S)$ which is the parameter posterior. On the other hand, we already assumed that $\rho(D \mid \Theta_{ij}, S)$ has a multinomial distribution. Since the Dirichlet distribution is the conjugate for multinomial distribution, by assuming a Dirichlet prior for parameters the parameter posterior would also follow a Dirichlet distribution. Accordingly, we assume that $\rho(\Theta_{ij} \mid S)$ has a Dirichlet distribution and thus the posterior distribution, $\rho(\Theta_{ij} \mid D, S)$, is also Dirichlet. It is well known that the mean of the Dirichlet distribution for each of the variables is

$$E(\theta_{ijk} \mid D, S) = \frac{N'_{ijk} + N_{ijk}}{N'_{ij} + N_{ij}} \tag{10}$$

where N_{ijk} is the number of cases in dataset where $X_i = k$ and $Pa(X_i) = j$ and $N_{ij} = \sum_k N_{ijk}$. Similarly, N'_{ijk} is the prior Dirichlet hyperparameter which serves as a pseudo count. In other words, we can assume that there is an imaginary (equivalent) sample with size N' which reflects our belief in the prior distribution and N'_{ijk} is the number of cases in the imaginary dataset where $X_i = k$ and $Pa(X_i) = j$. By plugging in (10) into Eq. (9) and summing over l, we get an expression for $\rho(D \mid S)$. As discussed earlier we pick $\rho(D,S) = \rho(D \mid S)\rho(S)$ as the network score, thus

$$\rho(D,S) = \rho(S) \prod_i \prod_j \frac{\gamma(N'_{ij})}{\gamma(N'_{ij} + N_{ij})} \prod_k \frac{\gamma(N'_{ijk} + N_{ijk})}{\gamma(N'_{ijk})} \tag{11}$$

where γ is the *Gamma* function [30]. This is called **BD** metric which stands for Bayesian metric with Dirichlet priors. Here $\rho(S)$ refers to our prior belief regarding the network structure which is usually assumed to follow a uniform distribution. The BD metric is not score equivalent, i.e. it does not give the same score for different networks in the same equivalence class. As discussed earlier, an equivalence class is a set of networks that imply the same assertions of conditional independency. Thus, it is desirable that our metric assigns the same score to all the members of the same class. [30] shows that the only Dirichlet hyperparameters that give a score equivalent metric are

$$N'_{ijk} = N' \rho_0(X_i = k, pa(X_i) = j) \tag{12}$$

where ρ_0 is some prior joint probability distribution. Substituting (12) into the BD metric, we obtain a new metric called **BDe** metric. This represents a score equivalent Bayesian metric with Dirichlet priors. Typically the prior distribution ρ_0 is assumed to be uniform, i.e. $\rho_0(X_i = k, Pa(X_i) = j) = \frac{1}{r_i q_i}$. Consequently, the hyperparameters would be equal to

$$N'_{ijk} = \frac{N'}{r_i q_i} \tag{13}$$

where r_i is the number of different categories for variable X_i and q_i is the number of different parent configurations for the same variable. Hence if we sum the pseudo counts over all $r_i \times q_i$ entries in the CPT, we get a total equivalent sample size of N'. This prior is called the **BDeu** prior, where the "u" stands for uniform.

3.2. Probabilistic inference in BNs

Bayesian networks provide full representation of the joint probability distribution over their random variables. Thus, we can use them to answer any conditional probabilistic queries about the domain. This is called *probabilistic inference* and usually deals with inferring the state of a set of query variables given the state of some evidence (or observation) variables. *Probability propagation* and *belief updating* are also popular terms that are used in the literature for addressing this problem.

To answer these queries we need to compute a conditional probability distribution. We denote this by $p(Q \mid E)$ where Q and E are query and evidence subsets of variables respectively. We can use Bayes theorem to compute this distribution:

$$p(Q \mid E) = \frac{p(Q, E)}{p(E)} \tag{14}$$

Here, the joint distribution learned by the Bayesian network, which has a factorized expression as in (4), can be used to compute the joint distributions. In fact, we can use marginalization (summing out over unwanted variables) to compute any desired probability. However, this will take exponential time. Therefore, researchers have looked for more efficient algorithms to address this problem. Variable elimination algorithm [31] systematically marginalizes out the irrelevant variables to compute the desired marginal distribution. Junction tree algorithm, which is a general-purpose algorithm for computing conditional and marginals on graphs, is the other inference method that is used for BNs. In a very brief summary, the steps in junction tree algorithm are as follows: First, the BN is converted to a moral graph meaning that arc directions are removed and all co-parents of a common child are connected to each other. Then the moral graph is triangulated meaning that we add chords so that every cycle in the graph has a chord. The next step is to build a graph in which each node corresponds to each maximal clique in the triangulated graph. By definition, a clique is a fully connected subset of nodes in a graph. Finally we find the associated junction tree from this clique graph. The basic idea behind building the junction tree is that the probability distribution can be represented as a product of clique distributions. Having built the junction tree, we can use Pearl's message passing algorithm [21] to update our belief regarding the conditional and marginal distributions in the graph. The core idea is to use Bayes theorem to update clique distributions in a local manner. For a detailed discussion about this algorithm readers are referred to [32,33]. Although these exact inference algorithms improve the computation process, they are still very slow for problems with many variables. Therefore for more complex problems researchers resort to approximate techniques.

Sampling (Monte Carlo) methods are an important class of approximate techniques for inference in BNs. This is done by drawing a large number of random configurations over the random variables using the Bayesian network model. Then, any probabilistic quantity can be approximated by using these samples. Many different algorithms are proposed to perform this type of sampling. These algorithms propose different methods to generate the samples and also the way the probabilities are computed given the samples. Probabilistic logic sampling [34], likelihood weighting [35] and Gibbs sampling [36] are among the most popular methods in this context.

3.3. Conditional independence properties of a BN

The pure structure of a Bayesian network can be used to infer the conditional independencies between random variables. By definition X and Y are conditionally independent given Z if and only if:

$$X \perp Y|Z \iff p(X, Y|Z) = p(x|Z)p(Y|Z) \tag{15}$$

This means that if we observe Z, knowing Y will not provide us with any additional information about X. Inferring conditional independencies from a BN is based on the underlying idea that "dependence" is associated with "connectedness" in the graph and "independence" with "separation". In other words, two nodes X and Y are **d-separated** if all the paths between them are blocked. In finding a path, the direction of the arcs is not important. There are two main rules that determine if an existing undirected path between two nodes is blocked or not.

1. A so-called *collider* or *v-structure*, $s \rightarrow m \leftarrow t$, blocks a path unless m or one of its descendants is in Z.
2. A collider-free path is blocked by Z if any member of Z is present in the path.

The extension of the above definition to sets of variables is straightforward (i.e., two sets are separated if and only if each element in one set is separated from every element in the other). Using the d-separation rules we can obtain useful insights regarding the conditional independencies between variables. D-separation criteria has an important implication in the Bayesian networks. Each node is independent of all the other nodes given its **Markov blanket** (By definition, the Markov blanket of a node in a DAG consists of its parents, its children and other co-parents of its children). These expressions prove to be highly informative when we want to extract some knowledge about the conditional independencies encoded by a Bayesian network.

4. Results and discussion

It is desirable for decision makers to discover the structure and dependencies among multiple factors which form the behavior of a complex system and use them to forecast the future behavior of the system. Building such a model enhances the general insight about the system and facilitates decision making tasks. In this paper, we address the demand response issue in the context of Smart Grid technology. This is a complex system which comprises of multiple random components. Our goal is to discover the dependencies between the contributing factors and probabilistically quantify their impact on the electricity demand. We seek to develop predictive models at the local (each customer) and global (aggregate customers) scales. In this regard, we use Bayesian networks to learn the dependencies as well as quantifying the joint probability distribution.

In order to learn a Bayesian network structure we need to choose a scoring metric as well as a search strategy. In this regard, we pick BDeu score and use tabu search algorithm in searching for the optimal network. We initially start with some arbitrary

network structure and find its score. Then the score change for all legal arc operations, one at a time, is calculated. By arc operations we mean arc addition, arc deletion and arc reversal. Next, we pick the arc operation with the highest score gain and apply it. We continue this procedure until a desired convergence rate in score is satisfied. This algorithm is not guaranteed to find a global optimum but only a local optimal structure. To avoid this problem, we can repeat the procedure for multiple initial structures and pick the best one. We use *bnlearn* package from R repository to conduct the simulations [37,38].

Learning the Bayesian network structure and parameters fully defines the joint probability distribution. To evaluate the quality of the learned model we need to assess its goodness of fit for a training dataset. Moreover, we can use a test dataset to quantify the generalization power of the learned model when applied to new datasets. We consider 90% of the data as training and 10% as the test set. In our problem, we pick normalized root mean squared error (NRMSE) to evaluate the performance of the model in predicting demand. NRMSE is defined as:

$$NRMSE = 100\% * \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{y_{max} - y_{min}} \tag{16}$$

where y_i denotes the observed value for variable y, $\hat{y}_i$ represents the predicted value by the model, $y_{max} - y_{min}$ is the variable range and n is the number of observed samples. In our problem, we take $\hat{y}_i$ to be the mean of the learned probability distribution for node $demand_t$. Since we discretize the domain of $demand_t$ to some bins, the BN model finds a probability mass function over the bins. We use bin midpoints to compute the distribution mean and standard deviation.

4.1. Dataset description

We use the electricity consumption data collected by Pacific Northwest national lab (PNNL) in a project called "Olympic Peninsula". The data consists of smart meter readings, i.e. electricity consumption (Watts), for 82 customers in three zip codes in the state of Washington. The project was carried out between April 1st 2006 and March 31st 2007 with a resolution of 5 min in recording the meter readings [39]. The customers were divided into three categories based on the pricing policy that was assigned to them: real time price (RTP), time-of-use price (TOU) and fixed price groups. One of the primary goals of this paper is to model the effect of price variations on the demand pattern, thus we only study the behavior of RTP customers (25 households) who received price signals every 15 min. We process the data to design 3 experiments, namely:

1. Aggregated customers - hourly temporal granularity.
2. Aggregated customers - 15 min temporal granularity.
3. Individual households - hourly temporal granularity.

Granularity refers to the temporal scale that we pick for aggregating price and consumption data. In case of aggregated customers, we extract the following attributes from data: month of year, time index of day (based on 15 min or hourly time resolutions), outside temperature, electricity price, day of week, is it weekend or not, demand at the current time step, demand at previous time step and demand at two previous time steps. Additionally, when dealing with the problem at the households level we also include customer ID (which uniquely distinguishes each household), aggregated demand of all customers, usage base and usage variations. The last two attributes represent the mean and standard deviation of consumption data for a particular household. The underlying idea is that they are supposed to characterize the household's consumption behavior. For the sake of space saving, these variables are denoted as follows in the associated graphs: *month*, *timeindex*, *temperature*, *price*, *dayofweek*, *isweekend*, $demand_t$, $demand_{t-1}$, $demand_{t-2}$, C_{ID}, *agg-demand*, *use-avg* and *use-var*.

4.2. Tuning discretization parameters

It is evident that some of the involved variables in our current problem take continuous values and some other take discrete values. Since we would like to use a discrete Bayesian network as our predictive model, we need to discretize continuous variables. To do so we use Fayyad and Irani [40] discretization method which is a widely used technique in the machine learning community. This method partitions the intervals according to a criterion that is based on minimum description length principle. Since, this is a supervised method we need to introduce the class variable to the algorithm. The class variable needs to be discrete beforehand. In our problem, we take the demand at current time as the class variable and discretize it to k bins. Then we use Fayyad and Irani technique to discretize other continuous variables. We use three different discretization criteria, namely equal width intervals, equal frequency intervals and k-means clustering for discretizing the demand variable.

By increasing the discretiztaion granularity, we will be able to predict at a finer scale but this will be at the cost of more parameters to be estimated.When the sample size is not big enough, this increase in the number of parameters will degrade the accuracy of the model. Therefore, we should always be mindful to keep a good balance between number of parameters and the sample size. The greater the number of observations, the finer granularity of the variables permitted.

Table 1 illustrates the effect of discretization method and number of bins on the model error for 15 min and hourly time granularities.

The dicretization scheme may also influence the shape and smoothness of the predicted probability distribution. We show this effect in Fig. 2 where three BNs with different discretization schemes are used to predict $\rho(demand_t|price > 48)$. We use junction tree algorithm from *gRain* package to infer any desired (conditional) probability distribution using the learned BN model [41]. The resulting distribution would be a histogram over the bins. We have used kernel density estimation with Gaussian kernels to assign a continuous distribution to the learned histograms. Fig. 2 shows that the equal width discretization gives the smoothest pdf. It should be mentioned that data outliers should be removed before using equal width discretization method since they can dramatically influence the results. Hereafter, we pick the BN obtained

Table 1
NRMSE(%) in predicting aggregated demand using discrete Bayesian network.

	Cluster			Equal width			Equal frequency		
	5	30	50	5	30	50	5	30	50
Hourly	7.90	7.20	7.22	7.27	7.29	7.25	9.37	7.31	7.45
15-min	5.94	4.91	4.97	5.23	4.91	4.94	9.46	5.28	5.11

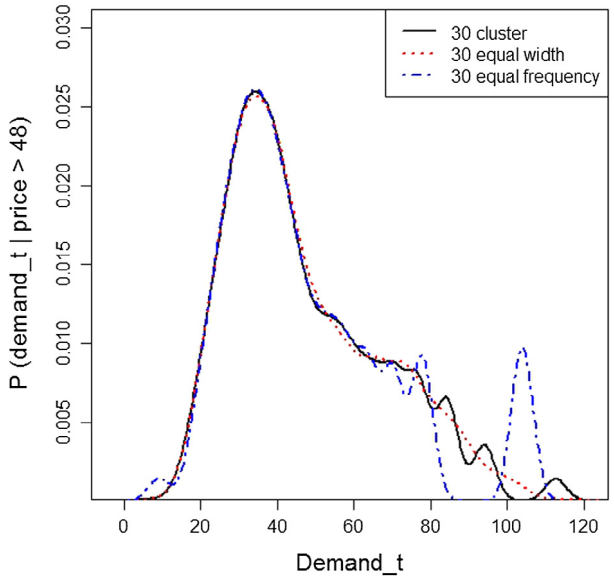

Fig. 2. Effect of discretization scheme on predicting hourly aggregated demand.

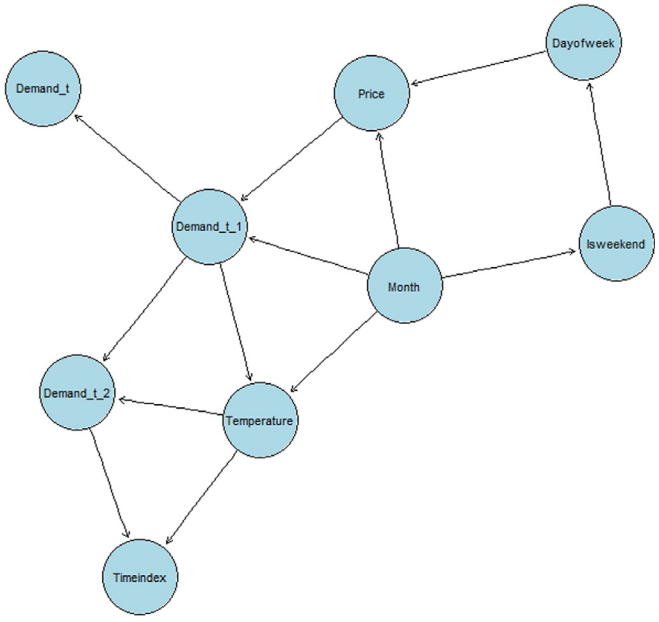

Fig. 3. Learned Bayesian network structure for modeling aggregated demand – hourly granularity.

from discretizing the demand to 30 equal width intervals for further analysis while we emphasize that this is a modeling choice.

4.3. Modeling aggregated demand - hourly granularity

The learned BN structures for hourly time scale using 30 equal width intervals to discretize $demand_t$ is depicted in Fig. 3. As discussed in Section 3.3, the pure structure of the learned BN can be used to infer conditional dependencies and independencies among the variables. The Markov blanket of $demand_t$ is $demand_{t-1}$, which means that demand at the previous time step can be used to fully specify the demand at a desired time step. The learned model can be used to answer any probabilistic query in the domain. For instance, we can forecast aggregated demand at any time conditional on any desired subset of variables that are observed.

Fig. 4 illustrates expected value of demand given a particular month and temperature and compares it versus the observed demand and unconditional demand mean. The unique modeling capability of Bayesian networks allows us to conduct any conditional inference even if only a subset of variables are observed. In other words, even if only one predictor variable is observed we can still get a reliable prediction for the target variable. However, most of the traditional forecasting methods need a complete instantiation of the predictor variables in order to compute a prediction for the target dependent variable.

4.3.1. Gaussian versus discrete Bayesian networks

Discretization is one way to deal with continuous variables in BNs. The other most widely used model to deal with continuous variables is to use the so-called *Gaussian Bayesian networks* (GBN). Here, we assume that each variable has a Gaussian distribution where its mean is a linear combination of its parents' values and its standard deviation is constant. It is shown [42] that given these local distributions the global joint pdf over all random variables would be a multivariate normal distribution. Similar to discrete Bayesian networks a Bayesian scoring metric can be used in searching for the best BN model when the structure is unknown. This score is called **BGe** which represents a Bayesian and score equivalent metric for Gaussian networks. For a detailed discussion about GBNs and the Gaussian score refer to [42]. In this research we have also considered the Gaussian BN model. The idea is that discretizing continuous variables (even with fine resolution) leads to some information loss while using the continuous distributions eliminates this problem. However, by discretizing the variables in the BN we are not imposing any particular distribution to the data and the approach is totally data-driven. On the contrary, in Gaussian BNs we are imposing a strong assumption on the variables,

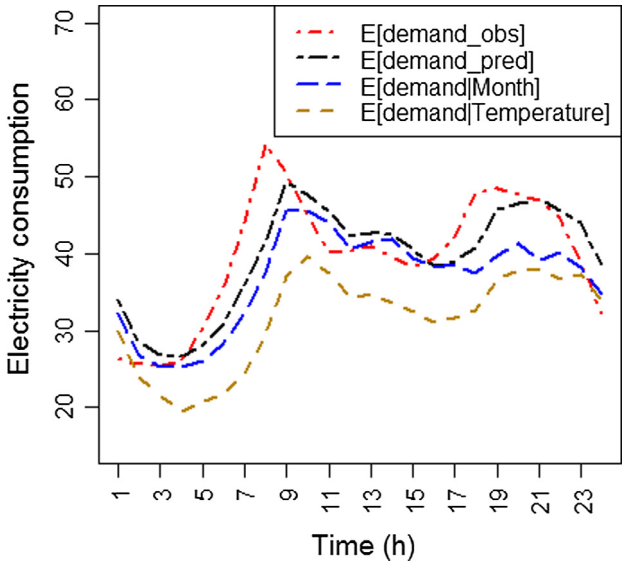

Fig. 4. Expected value of aggregated demand with hourly granularity-comparing the observed mean versus the conditional and unconditional predicted mean.

Table 2
NRMSE(%) in predicting aggregated demand by discrete and Gaussian Bayesian networks – hourly granularity.

	Gaussian BN	discrete BN-6 variables
Hourly	6.97	7.29
15-min	4.77	4.91

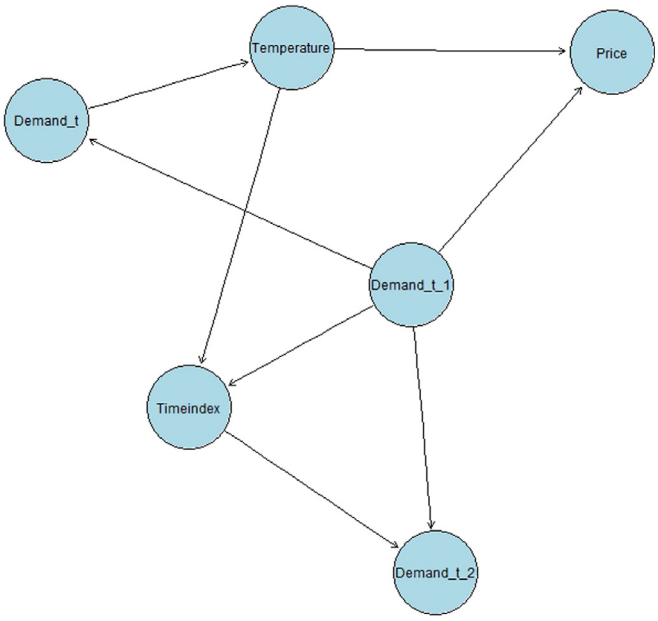

Fig. 5. Learned discrete Bayesian network structure for modeling aggregated demand – hourly granularity.

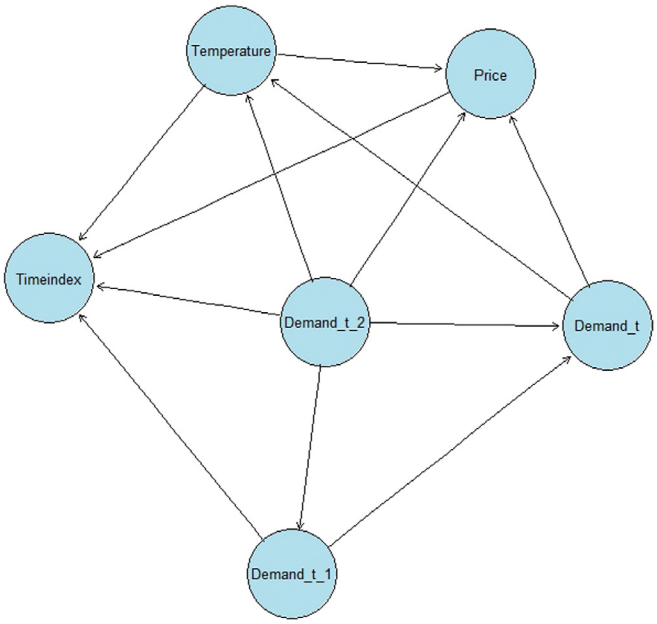

Fig. 6. Learned Gaussian Bayesian network structure for modeling aggregated demand – hourly granularity.

that they follow a Gaussian distribution. In practical applications this assumption almost always deviates from reality but still it might be a good choice depending on the specific data.

In this work, the original data consists of both discrete and continuous variables (hybrid problem). In order to apply the Gaussian BN we disregard three discrete variables namely *month*, *dayofweek* and *isweekend* since assuming a Gaussian distribution for them seems meaningless. Then, we learn a Gaussian and also a discrete BN for the remaining 6 variables. As before, we use equal width discretization with 30 bins for learning the discrete BN. The NRMSE for these two models is tabulated in Table 2. Based on Table 2 we can observe that the error of predicting demand is lower when using the Gaussian BN, however the improvement is not significant. Figs. 5 and 6 compare the structure of the two learned BNs for hourly time scale. We can see that these two graphs imply different conditional independency assertions. For instance the Markov blanket of $demand_t$ in discrete BN model is consisted of $demand_{t-1}, temperature$ while in the Gaussian BN it is $demand_{t-1}, demand_{t-2}, price, temperature$. Besides, the *price* and $demand_t$ are directly dependent in the Gaussian BN while in the discrete BN they are dependent through the *temperature* or $demand_{t-1}$.

To get more insight regarding the differences of the two models, we compute the probability distribution of an arbitrary conditional query using the two models. Fig. 7 illustrates the unconditional distribution of $demand_t$ as well as the conditional distribution $\rho(demand_t|price > 48)$ using the discrete and Gaussian BNs. It is observed that the general shape of the conditional pdf obtained from the discrete model is more similar to the unconditional distribution. This result is somewhat expected, since the discrete BN is a nonparametric data-driven model that tries to capture the true distribution while the Gaussian BN always assigns a Gaussian pdf to the variables.

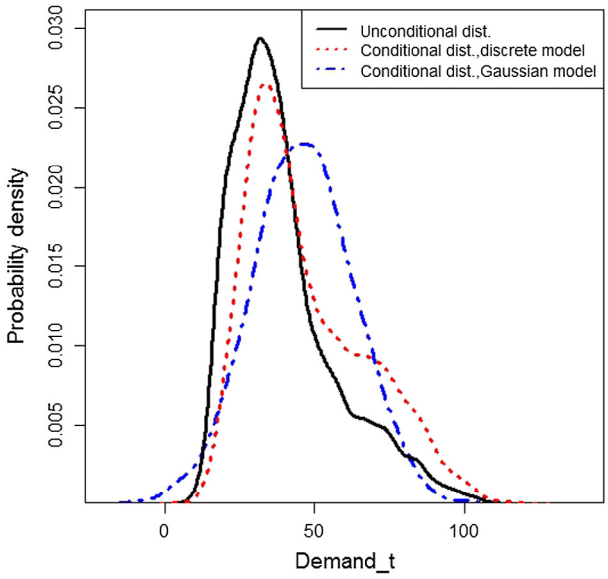

Fig. 7. Comparison of conditional probability distribution of hourly aggregated demand given price, $\rho(demand_t|price > 48)$, using Gaussian and discrete BNs.

4.4. Modeling aggregated demand - 15 min granularity

The learned BN structure for 15 min granularity using 30 equal width intervals to discretize $demand_t$ is depicted in Fig. 8. Comparing Figs. 8 and 3, we observe that the learned BN structure for demand modeling at hourly and 15 min granularities are different. This is reasonable, since data at various time scales might show different hidden structures and this is in fact a question that has been tried to be answered in this paper. For instance, aggregating data from 15 min to 1 h may cause us to lose some fine scale variations and thus discover new dependency structures in the data. Nevertheless, many important dependency relations are still unchanged. For example, $demand_t$ is directly dependent to $demand_{t-1}$ in both structures. Also, *price* variable is directly dependent on *dayofweek*, *month* and $demand_{t-1}$ in both models. It is clear from the present analysis that the dependence structure in the data depends on the time resolution at which the data is observed. Training and calibration of predictive models should therefore be adapted to specific resolutions. Besides, BN models are defined by their structure

and their estimated parameters (not shown in the figures) and their reliability is validated through tables and graphs reporting on NRMSE and confidence intervals which are presented throughout the paper.

We can use the learned model to predict the system's behavior for the case of unobserved situations. We use the test datset to demonstrate the generalization power of the model. Fig. 9 illustrates the model results for predicting demand using 50 random samples from the test dataset for 15 min time scale. Here, the black points show the predicted mean and the blue bars represent 2σ uncertainty bound around the mean. The red points represent the actual observed value for the demand. We can see that almost all of the observed values fall within the predicted range.

4.5. Modeling individual demand - hourly granularity

The advent of smart meters enables decision makers to set individualized policies for each particular customer in the electricity market. By studying the consumption behavior of each household, the strategists are able to target certain groups of customers and enforce appropriate policies to them in order to shape the market as needed. The first step in designing individualized policies is to have a predictive model that can envision the expected behavior of the customers in response to the implemented policies. The traditional regression models used in predicting the demand usually are not successful in predicting at the households scale. They will need too may parameters to tune and often get too complicated to handle easily. Here, we propose the use of Bayesian networks to predict demand at the households level and also to capture the dependencies between the involved variables.

The dataset used to train the model has 139,277 samples which consists of the meter readings for 25 customers over the course of a year as discussed earlier. We use hourly scale to aggregate the price and consumption data. Thus each sample represents the consumption of one particular customer at some specific time subject to some price and temperature values. We use 30 equal width intervals to discretize the demand and then Fayyad and Irani algorithm is applied to the rest of the continuous variables. The learned BN is depicted in Fig. 10. The NRMSE in predicting demand using this model is 11.82%. This error is relatively higher than the error of predicting the aggregated demand for all customers. This is reasonable, since the demand has much more variations at the household's scale while in general aggregating smooths the behavior. The Markov blanket of $demand_t$ in this problem is consisted of $demand_{t-1}$ and *use-avg*. The average of usage is a variable that reflects the customer's consumption pattern.

Fig. 11 illustrates the predicted mean and an error bound in predicting $demand_t$ for 50 samples picked randomly from the test dataset using the learned BN. The black line represents the mean of the predicted distribution and the shaded area corresponds to

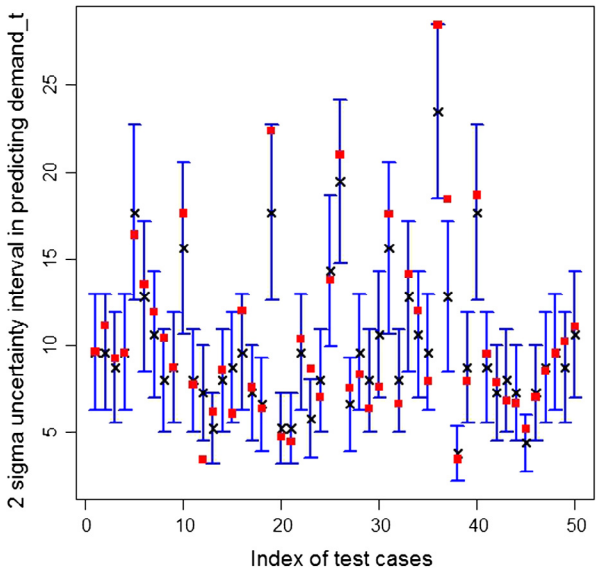

Fig. 9. Uncertainty interval in predicting aggregated demand with 15 min granularity-red dots represent the observed demand, the black points are the predicted mean and the blue bars correspond to 2σ interval around the mean.

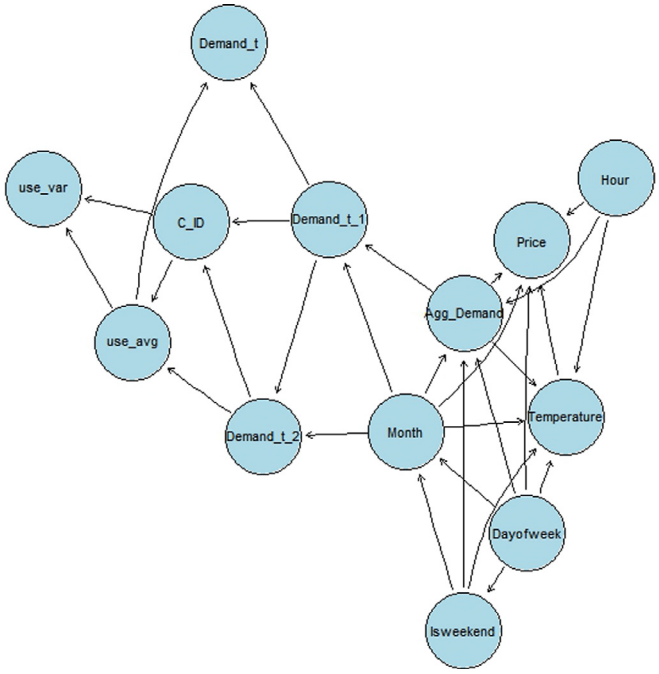

Fig. 10. Learned Bayesian network structure for predicting individual demand – hourly granularity.

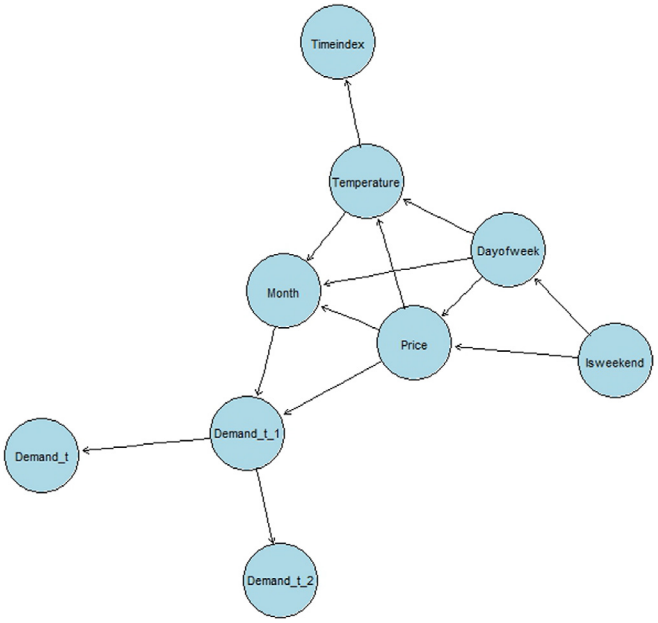

Fig. 8. Learned Bayesian network structure for modeling aggregated demand – 15 min granularity.

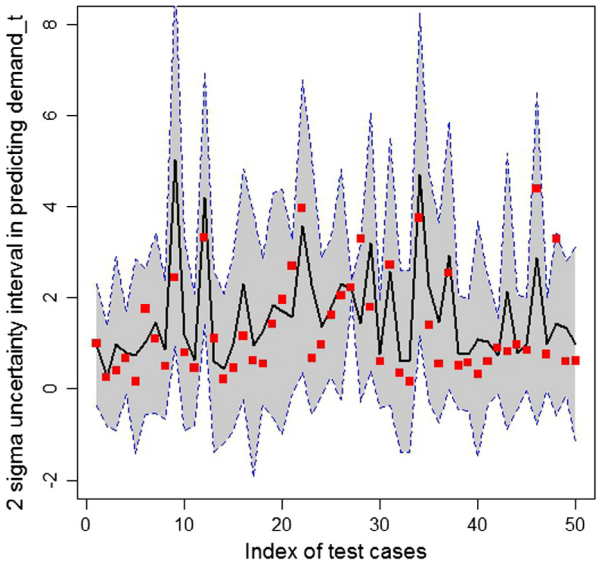

Fig. 11. Uncertainty interval in predicting individual demand with hourly granularity-red dots represent the observed demand, the black line is the predicted mean and the shaded area corresponds to 2σ interval around the mean.

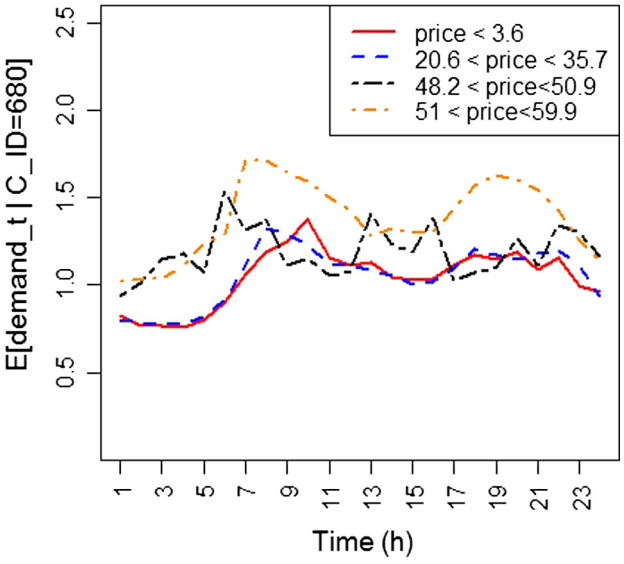

Fig. 13. Hourly variations of expected value of demand for customer number 680 for 4 different price ranges.

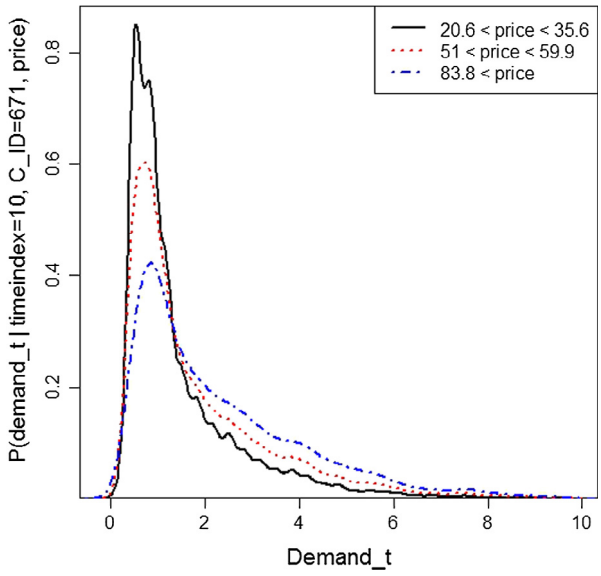

Fig. 12. Probability distribution of demand for customer number 671 at 10 am for 3 different price ranges.

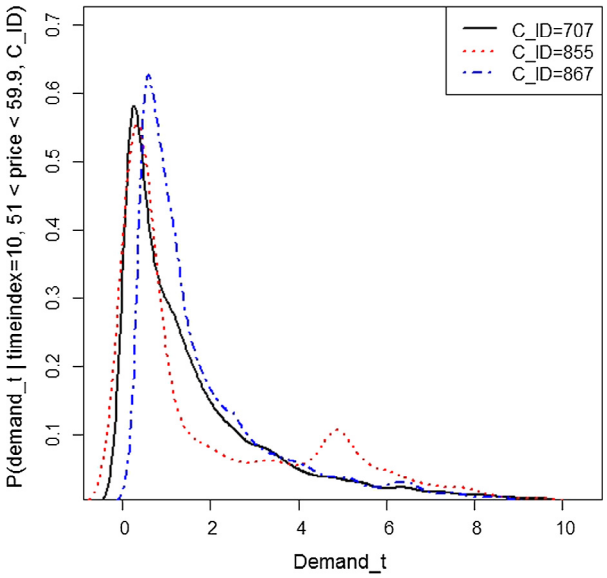

Fig. 14. Probability distribution of demand for customer IDs 707, 855, and 867, at 10am given that $51 < price < 59$ \$/MW h.

2σ deviation from the mean. Additionally, the red dots denote the actual observed value of $demand_t$ for each sample of the test dataset. The figure shows that the majority of the red points fall within the error bound limits and many of them are close to the mean. This suggests that the model is pretty successful in predicting the demand for unseen cases.

Using the learned BN model, we are able to predict the probability distribution of any desired variable conditional on any set of observed variables. For instance we can compute $\rho(demand_t|timeindex, C_{ID}, price)$ to get the probability distribution of demand at a specific time for a particular customer given the price. Here C_{ID} is treated as a random variable that uniquely distinguishes each individual household.

Assume the utility company is interested in testing the effect of price variations on consumption behavior of a target household at a particular time. Fig. 12 probabilistically answers this question for the utility company by quantifying $\rho(demand_t|timeindex = 10, C_{ID} = 671, price)$ for customer number 671. The variations of demand distribution with regard to different price intervals are clearly quantified. A strong trend is observed in this figure as the unit price is raised with a marked decrease in low-level demand, matched by an increase in high-level demand. This may be attributed to a strong correlation between external effects (such as temperature) and demand paired with a pricing mechanism that anticipates this correlation. Thus the low-level demand is not associated with external effects and was thus reduced as price was increased. The high level demand, on the other hand, could be associated with external effects and its rise coincides with an increase in price to mitigate peak usage (eg. the high-level rise might have been more pronounced without such a price increase).

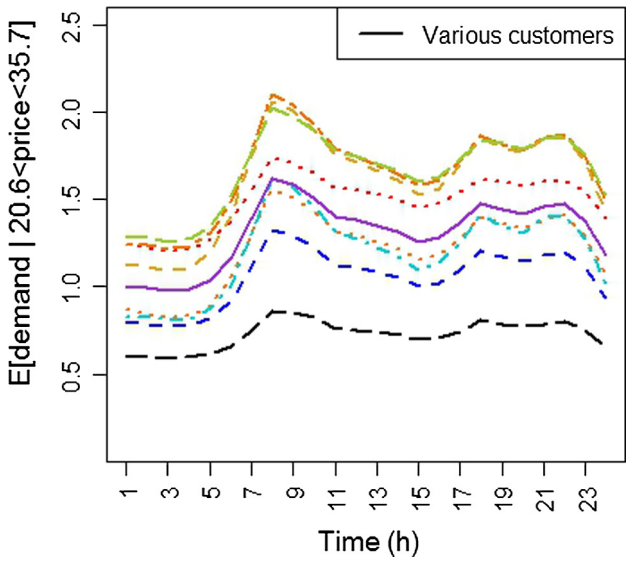

Fig. 15. Hourly variations of expected value of demand for 9 different customers given that $20.6 < price < 35.7$ \$/MW h.

Similarly, Fig. 13 illustrates hourly variations of the predicted mean of demand for a particular customer subject to various price intervals. We can see that the household's daily consumption profile has been altered due to the price variations.

As another application, we demonstrate the capability of the model in differentiating various categories of customers based on their response sensitivity to price incentives. The probability distribution of demand given the price value for 3 different customers is depicted in Fig. 14. This graph quantifies different behaviors of customers subject to a particular price. Fig. 15, similarly, depicts the temporal variations of predicted mean of demand for 9 customers in a day. These consumption profiles show the different consumption patterns of different households. These results can be used to design targeted pricing policies for different households while effectively capture the uncertainties in predictions. Moreover, it is observed that the demand of individual customers may deviate significantly from aggregated demand, and the ability of the present BN model to anticipate both types of demand demonstrates its value for a wide range of potential users.

5. Conclusion

In this paper, we attempt to provide solutions to some challenges in Smart Grids from a stochastic and data-driven point of view to help decision makers make more informed decisions. The boom of data streams in energy industry necessitates system modelers to utilize this data in order to improve the system performance. In this research, we introduced a framework to conduct multiscale probabilistic queries on electricity consumption data provided by Pacific Northwest National Lab.

To be more specific, we study the problem of probabilistic sensor based residential load forecasting. Multiple stochastic variables that shape the customer consumption behavior pose a challenge to this problem. We propose using Bayesian network, which is able to estimate the joint probability distribution in a complex system, as the framework for capturing the uncertainties and conducting probabilistic inferences. We also study the forecasting problem at the aggregated and individualized scales. The availability of the usage data at the households level provides a unique opportunity for modeling households consumption behavior which can lead to targeted pricing policies for each household. The results show that our model can assist in quantifying the confidence in predictions at the aggregate and personalized scales. Also the model can be used for demand response policies in order to assess the effect of pricing policies on the consumption behavior of the customers.

References

[1] Taylor JW, De Menezes LM, McSharry PE. A comparison of univariate methods for forecasting electricity demand up to a day ahead. Int J Forecast 2006;22:1–16.
[2] Hagan MT, Behr SM. The time series approach to short term load forecasting. IEEE Trans Power Syst 1987;2:785–91.
[3] Hosking J, Natarajan R, Ghosh S, Subramanian S, Zhang X. Short-term forecasting of the daily load curve for residential electricity usage in the smart grid. Appl Stochas Models Business Indus 2013;29:604–20.
[4] Edwards RE, New J, Parker LE. Predicting future hourly residential electrical consumption: a machine learning case study. Energy Build 2012;49:591–603.
[5] Yun Z, Quan Z, Caixin S, Shaolan L, Yuming L, Yang S. Rbf neural network and anfis-based short-term load forecasting approach in real-time price environment. IEEE Trans Power Syst 2008;23:853–8.
[6] Alfares HK, Nazeeruddin M. Electric load forecasting: literature survey and classification of methods. Int J Syst Sci 2002;33:23–34.
[7] Edwards RE, New J, Parker LE. Predicting future hourly residential electrical consumption: a machine learning case study. Energy Build 2012;49:591–603.
[8] Wong SL, Wan KK, Lam TN. Artificial neural networks for energy analysis of office buildings with daylighting. Appl Energy 2010;87:551–7.
[9] Kialashaki A, Reisel JR. Modeling of the energy demand of the residential sector in the united states using regression models and artificial neural networks. Appl Energy 2013;108:271–80.
[10] Ghasemi A, Shayeghi H, Moradzadeh M, Nooshyar M. A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management. Appl Energy 2016;177:40–59.
[11] Li Q, Meng Q, Cai J, Yoshino H, Mochida A. Applying support vector machine to predict hourly cooling load in the building. Appl Energy 2009;86:2249–56.
[12] Li K, Su H, Chu J. Forecasting building energy consumption using neural networks and hybrid neuro-fuzzy system: a comparative study. Energy Build 2011;43:2893–9.
[13] Sevlian R, Rajagopal R. Short term electricity load forecasting on varying levels of aggregation; 2014. arXiv:1404.0058.
[14] Jain RK, Smith KM, Culligan PJ, Taylor JE. Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of temporal and spatial monitoring granularity on performance accuracy. Appl Energy 2014;123:168–78.
[15] Chehouri A, Younes R, Ilinca A, Perron J. Review of performance optimization techniques applied to wind turbines. Appl Energy 2015;142:361–88.
[16] Hong T, Fan S. Probabilistic electric load forecasting: a tutorial review. Int J Forecast 2016;32:914–38.
[17] Mori H, Ohmi M. Probabilistic short-term load forecasting with gaussian processes. In: Proceedings of the 13th international conference on, intelligent systems application to power systems. IEEE; 2005. 6 pp.
[18] Fan S, Hyndman RJ. Short-term load forecasting based on a semi-parametric additive model. IEEE Trans Power Syst 2012;27:134–41.
[19] Bracale A, Caramia P, Carpinelli G, Di Fazio AR, Varilone P. A bayesian-based approach for a short-term steady-state forecast of a smart grid. IEEE Trans Smart Grid 2013;4:1760–71.
[20] Vlachopoulou M, Chin G, Fuller JC, Lu S, Kalsi K. Model for aggregated water heater load using dynamic bayesian networks; 2012.
[21] Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufman; 2014.
[22] Heckerman D. A tutorial on learning with Bayesian networks. Springer; 1998.
[23] Langseth H, Portinale L. Bayesian networks in reliability. Reliab Eng Syst Safety 2007;92:92–108.
[24] Cormen TH. Introduction to algorithms; 2009.
[25] Murphy KP. Machine learning: a probabilistic perspective. MIT press; 2012.
[26] Nielsen TD, Jensen FV. Bayesian networks and decision graphs. Springer Science & Business Media; 2009.
[27] Lam W, Bacchus F. Learning Bayesian belief networks: an approach based on the mdl principle. Comput Intell 1994;10:269–93.
[28] Friedman N, Goldszmidt M. Learning Bayesian networks with local structure. In: Learning in graphical models. Springer; 1998. p. 421–59.
[29] Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Machine Learn 1992;9:309–47.
[30] Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learn 1995;20:197–243.
[31] Zhang NL, Poole D. Exploiting causal independence in Bayesian network inference. JAIR 1996;5:301–28.
[32] Shafer GR, Shenoy PP. Probability propagation. Ann Math Artif Intell 1990;2:327–51.
[33] Lauritzen SL, Spiegelhalter DJ. Local computations with probabilities on graphical structures and their application to expert systems. J Roy Stat Soc, Ser B (Methodol) 1988:157–224.
[34] Henrion M. Propagation of uncertainty by probabilistic logic sampling in Bayes networks. In: Uncertainty in artificial intelligence, vol. 2; 1988. p. 149–164.

[35] Fung R, Chang K-C. Weighing and integrating evidence for stochastic simulation in Bayesian networks; 2013. arXiv:1304.1504.
[36] Gilks WR, Thomas A, Spiegelhalter DJ. A language and program for complex Bayesian modelling. Statistician 1994:169–77.
[37] Scutari M. Learning Bayesian networks with the bnlearn R package. J Stat Softw 2010;35:1–22.
[38] Nagarajan R, Scutari M, Lèbre S. Bayesian networks in R with applications in systems biology, vol. 48. Springer; 2013.
[39] Chassin DP. Olympic peninsula demonstration testbed results. Technical report. Pacific Northwest National Laboratory; 2010.
[40] Fayyad UM, Irani KB. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th international joint conference on artificial intelligence. Morgan Kaufmann Publishers Inc.; 1993. p. 1022–9.
[41] Højsgaard S. Graphical independence networks with the gRain package for R. J Stat Softw 2012;46:1–26.
[42] Geiger D, Heckerman D. Learning Gaussian networks. In: Proceedings of the tenth international conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 1994. p. 235–43.