Predicting Future Hourly Residential Electrical Consumption: A Machine Learning Case Study

Richard E. Edwards[a], Joshua New[b], Lynne E. Parker[a]

[a]*Department of Electrical Engineering and Computer Science*
University of Tennessee, Knoxville TN, USA
(redwar15,parker)@eecs.utk.edu
[b]*Building Envelopes Research Group*
Oak Ridge National Laboratory, Oak Ridge TN, USA
newjr@ornl.gov

Abstract

Traditional whole building energy modeling suffers from several factors, including the large number of inputs required to characterize the building, the specificity required to accurately model building materials and components, simplifying assumptions made by underlying simulation algorithms, and the gap between the as-designed and as-built building. Prior work has attempted to mitigate these problems by using sensor-based machine learning approaches to statistically model energy consumption. We refer to this approach as sensor-based energy modeling (sBEM). However, a majority of the prior sBEM work focuses only on commercial buildings. The sBEM work that focuses on modeling residential buildings primarily focuses on monthly electrical consumption, while commercial sensor-based models focus on hourly consumption. This means there is not a clear indicator of which machine learning approach best predicts next hour residential consumption, since these methods are only evaluated using low-resolution data. We address this issue by testing seven different machine learning algorithms on a unique residential data set, which contains 140 different sensors measurements, collected every 15 minutes. In addition, we validate each learner's correctness on the ASHRAE Great Energy Prediction Shootout, using the original competition metrics. Our validation results confirm existing conclusions that Neural Network-based methods perform best on commercial buildings. However, the results from testing on our residential data set show that Feed Forward Neural Networks (FFNN), Support Vector Regression (SVR), and Linear Regression methods perform poorly, and that Least Squares Support Vector

Machines (LS-SVM) perform best – a technique not previously applied to this domain.

Keywords: energy modeling, sensors, machine learning

---

1. Introduction

Residential and commercial buildings consitute the largest sector of U.S. primary energy consumption at 40% (U.S. Department of Energy, 2010). Building energy efficiency is often described as the "low hanging fruit" for reducing this consumption and the requisite greenhouse gas emissions. Building energy modeling is a crucial tool in the development of informed decisions regarding the augmentation of new and existing buildings. Whole building energy modeling is currently utilized for several purposes: identifying energy consumption trade-offs in the building design process, sizing components (e.g., HVAC) for a specific building, optimizing control systems and strategies for a building, determining cost-effective retrofit packages for existing buildings, and developing effective building codes, tax/rebate incentives and Research, Development, Demonstration, and Deployment (RDD&D) roadmap activities required to meet energy reduction goals set by numerous organizations, utility companies deferring infrastructure upgrades, and local/state/federal governments.

There are two general types of energy modeling: traditional "forward" modeling and "inverse" modeling. Most energy modeling software are "forward" models, which utilize input parameters such as weather data, building geometry, envelope composition with material properties (e.g., thermal conductivity, specific heat), equipment systems with component properties, and operating schedules. The software then uses an engineering model to quickly step forward through simulated time in order to calculate the energy consumption of the specified building. There are hundreds of these software packages available; twenty of the most popular programs, including the world's most popular DOE-2 and the next-generation code EnergyPlus, have been contrasted previously (Crawley et al., 2008).

Traditional "inverse" modeling, on the other hand, assumes a known mathematical relationship between a set of inputs (e.g., outdoor temperature) and an output (e.g., electrical consumption). Using the assumed relationship, "inverse" modeling uses statistical methods, such as Linear Regression, to solve for the best model parameters. The relationship between the

inputs and output is generally based on engineering domain knowledge. For example, the Three-parameter Cooling model (3PC) and Three-parameter Heating model (3PH) assume that there exist a single change point where the electrical consumption trend changes with respect to outdoor temperature (Kissock et al., 2003). These types of traditional "inverse" models use domain knowledge to derive the mathematical relationship, rather than a data-driven approach where the data determines the model that best describes the relationship.

Sensor-based energy modeling can be viewed as a hybrid between "forward" modeling and "inverse" modeling approaches. This data-driven approach assumes that the sensor data provides a viable model for the entire building – the "forward" component. This means the sensor data encodes the state of weather, building envelope, equipment, and operation schedules over time. Through the application of machine learning algorithms, an approximation of the engineering model is derived statistically – the "inverse" component. However, the machine learning algorithms used by Sensor-based energy modeling allows the data to determine the best model, rather than engineering domain knowledge that may not always be applicable.

Sensor-based energy modeling serves as an alternative approach to traditional "forward" and "inverse" modeling. In fact, there are numerous sensor-based studies that focus on predicting current and future electrical consumption for commercial buildings (Kreider and Haberl, 1994; Karatasou et al., 2006; Li et al., 2011). In addition, these studies have established which machine learning techniques perform well at modeling commercial electrical consumption. However, very little sensor-based work focuses on modeling electrical consumption for residential buildings, rather than commercial buildings. In fact, most sensor-based studies conducted with residential buildings model monthly electrical consumption (Kolter and Ferreira Jr, 2011), while commercial building studies model hourly consumption. This means the few established methods for residential buildings are only tested and verified on monthly data. Therefore, there is a need to explore additional techniques on higher granularity data sets and to establish which machine learning techniques truly perform best at modeling residential electrical consumption.

The gap between the residential and commercial studies stems from the fact that residential data sets lack granularity and are generally collected from monthly utility statements. In this work, we narrow the gap between these studies by exploring seven different machine learning techniques, and

determining which ones are best for predicting next *hour* electrical consumption within residential buildings. We achieve this by using a new residential data set, leveraging the proven methods from the literature for commercial buildings, and introducing new techniques that have not been previously applied to this domain.

The remainder of the paper is organized as follows: Section 2 compares traditional energy modeling and sensor-based statistical modeling; Section 3 discusses related work in the area of sensor-based machine learning applied to building energy modeling; Section 4 provides a technical overview of the different machine learning algorithms we explore within this work; Section 5 presents a detailed description of the residential data set, experimental design, and evaluation criteria; Section 6 presents results for predicting next hour residential electrical consumption, as well as results that validate the machine learning algorithms' correctness; Section 7 provides discussion about the results; and Section 8 presents our conclusions and future directions.

2. Traditional Modeling vs Sensor-Based Modeling

Both forward and inverse modeling approaches, individually, suffer from several problems that are mitigated, if not solved, through sensor-based energy modeling. First, very few design firms have the expertise and can absorb the time and cost necessary to develop a thorough set of inputs during the design phase of a building. Most do so primarily for the largest of projects, despite the fact that the most important energy-consuming decisions are made during this phase and are least costly to remedy during early design. While sensor-based energy modeling does require existing sensor data, and thus implies an existing building, machine learning software trained on data from a similar reference building can function as an approximation engine and may provide sufficiently accurate results for quick feedback during early, iterative building design. Second, there is always a gap between the as-designed and as-built building. During construction, changes are made out of necessity, convenience, or negligence (e.g., lack of insulation in a corner), and many changes are very rarely communicated to designers or energy modelers. Sensors obviously eliminate this problem by measuring the actual state of the building rather than a designer's intentions. Third, sufficient knowledge is rarely available to accurately classify the dynamic specificities of equipment or a given material. Most energy modelers use the *ASHRAE Handbook of Fundamentals* (American Society of Heat-

ing, Refrigerating and Air-Conditioning Engineers, Inc., 2009) to estimate thermal and related properties based on typical values. Many others use the manufacturer's label information when available. However, few modelers put the materials and equipment through controlled laboratory conditions, or the appropriate ASTM test method, to determine properties of the specimen actually used in the building. The sensor-driven approach can not only capture the current/actual performance of the material, but also its degradation over time. Fourth, traditional modeling approaches can involve manually defining thousands of variables to codify an existing building. Since multiple experts may encode a specific building in many different ways, the large required input space lends itself to problems with reliability/repeatability and ultimately validity. Sensors are much more reliable and repeatable in reporting measured data over time, until a sensor or data acquisition system fails. Fifth, both the inverse statistical model and forward engineering models, by their very nature, necessarily require fixed assumptions and algorithmic approximations. Machine learning allows asymptotic approximation to the "true" model of the data, limited solely by the amount or quality of data provided, the capabilities of the algorithm utilized, or the time available to compute/learn from the available data.

For all its advantages, sensor-based energy modeling also introduces some of its own concerns and limitations. First, the additional cost associated with acquisition and deployment of sensors is not required by previous modeling approaches. Sensor development and costs are dropping with the same transistor density doubling every 18 months as defined by Moore's Law (Schaller, 1997). Increasingly sophisticated peel-and-stick, wireless mesh, energy-harvesting, system-on-a-chip sensors are becoming readily available. While the increase in capabilities and reduction in costs continue, it is currently infeasible to heavily instrument a building cost-effectively. Second, the number, type, and placement of sensors required to sufficiently capture the state of different building types is an open question. While this paper does not address this problem, selecting an optimal or approximate subset of sensors for energy modeling for a particular building could affect the cost of future buildings, but extrapolation across building types is an open problem; sensor counts/types could vary based upon the metric(s) being predicted. It is anticipated that shared, web-enabled databases of heavily instrumented buildings will help resolve this current issue. Third, sensors, data acquisition systems, and the physical infrastructure upon which they rely can be unstable and result in missing or corrupted sensor data values.

To mitigate this real-world issue, intelligent quality assurance and control algorithms (Ibargüengoytia et al., 2001) can be applied to detect and/or correct corrupted sensor values. The sensor pre-processing system we currently use notifies assigned personnel via email messages for data channels exhibiting out-of-range errors, using simple statistical tests. Lastly, determining the best machine learning algorithm for a given learning task is an open question. While there exist taxonomies for classifying problem types and appropriate machine learning algorithms (Russell and Norvig, 2010), rarely is there a known algorithm that is best for solving a given problem (e.g., predicting next hour energy usage). This issue is mitigated by exploring seven different machine learning algorithms, and determining which algorithm or algorithms perform best.

3. Related Work

Many researchers have explored machine learning alternatives for modeling electrical consumption, both within commercial buildings and residential buildings. However, a majority of the studies have focused on commercial buildings. A notable study that used commercial building data is the Building Energy Predictor Shootout hosted by ASHRAE. The competition called for participants to predict hourly whole building electrical (wbe) consumption for an unknown building using environmental sensors and user-defined domain knowledge. The competition provided 150 competitors with data from September 1, 1989 until December 31, 1989 as training data, as well as testing data that had the target variables removed. Six winners were selected from the submitted predictions (Kreider and Haberl, 1994).

The overall winner, MacKay et al. (1994), used a Feed Forward Neural Network (FFNN) with Auto Relevance Detection (ARD). The author was not sure which inputs or variables were most beneficial for predicting the specified targets. Therefore, the author devised a method for exploring a wide variety of different inputs that would minimize the error caused by irrelevant inputs. This Auto Relevance Detection process drives the weights for irrelevant inputs toward zero and prevents the weights for other inputs from growing too large or overpowering the solution. This is achieved by reformulating weight regularization to obey a probabilistic model, where all parameters follow prior distributions and the weights are inferred using Bayesian inference. The results presented from this prior work provide strong incentive for exploring how effective FFNNs are at predicting future residential electrical

consumption. Our use of this method is discussed in more detail in Section 4.2.

Another winner used Piecewise Linear Regression (Iijima et al., 1994). The authors created three different linear functions for predicting wbe. The first model is dedicated to workdays, the second is dedicated to weekends, and the third is dedicated to modeling holidays. These models were combined using the provided temporal information: day, month, year, and hour. However, the method used in this work requires explicit temporal domain knowledge about the particular application area. Given that we lack such temporal domain knowledge for residential domains, we decided to explore an automated Piecewise Linear Regression process. We apply Hierarchical Mixture of Experts (HME) with Linear Regression, because it uses the training data to automatically build and integrate multiple linear models. Section 4.1 briefly describes Linear Regression, and Section 4.5 discusses HME with Linear Regression in greater detail.

Feuston and Thurtell (1994) used an ensemble of FFNNs, which involved training multiple FFNNs and combining them by averaging their predictions. The predictions for each FFNN were equally weighted and the networks were trained using the same training data, and possibly different initializations. This method assumes that all FFNN responses are equally important, which may harm or not improve accuracy over a single network. This can harm accuracy, especially if a majority of the FFNNs learn the same errors, and only a few networks learn to correct those errors. Therefore, we decided to explore a more balanced and general method for mixing multiple FFNNs. The HME approach, which we previously mentioned, combined with FFNN Experts, accomplishes the same task, except the predictions are combined based on the likelihood that each network produces the correct prediction; Section 4.5 provides more details.

A more recent wbe prediction study with commercial buildings uses Support Vector Machines (SVM) to predict monthly consumption (Dong et al., 2005). SVMs are built on the principle that minimizing structural risk produces a general model. In addition, SVMs have a proven upper bound on the error rate for classification problems (Vapnik, 1999). While we do not know of a proven upper bound for regression problems, minimizing structural risk can still produce general models. The results from this prior work and the known benefits from SVMs lead us to the application of Support Vector Regression (SVR), which is SVM adapted for Regression (Section 4.3). These prior results also encouraged us to explore an SVM variant, called

Least Squares Support Vector Machine (LS-SVM) (Section 4.4).

Karatasou et al. (2006) builds upon the success found with FFNN and explores selecting the most important inputs and network structure for the Building Energy Predictor Shootout data. In addition, the work explores another commercial building data set. The authors present impressive results on both buildings and out-performed the Shootout winner. However, the authors provide little discussion about what allowed them to obtain better performance or the key differences between other FFNN techniques. The results found within this study provide further incentive to explore the application of FFNN to predicting residential electrical consumption.

Another recent work, by Li et al. (2011), presents results for the Energy Predictor Shootout that are better than the overall winner as well. This approach uses an Adaptive Neuro Fuzzy Inference System (ANFIS), which deviates greatly from the previously published FFNN works. This method combines partitioning rules from Fuzzy Systems with the properties of FFNNs, which is similar to Fuzzy C-Means (FCM) with FFNN. However, the authors in this work fully use the Fuzzy Systems by using multiple partitioning functions, while the FCM with FFNN in our work uses a single partitioning function. Section 4.6 provides a more detailed description about FCM with FFNN.

These studies on commercial buildings provide insight into successful techniques, many of which have inspired several of the techniques we explore in this article. However, how successful are these techniques on residential buildings? The studies that involve residential buildings are generally conducted with monthly information collected from utility companies. This means that most residential studies do not provide hourly predictions, which is fairly different from our focus on predicting hourly wbe consumption. For instance, (Kolter and Ferreira Jr, 2011) focuses on modeling commercial and residential buildings, but all the whole building energy (wbe) measurements are only at a monthly resolution for all buildings. This restriction is created by the fact that utility companies measure residential electrical consumption at monthly intervals, while commercial electrical consumption is measured hourly.

Our research makes use of a new residential data set, called the Campbell Creek data set, which gives us a unique opportunity to predict next hour wbe electrical consumption for residential homes. The Campbell Creek data set contains approximately 140 different sensor measurements collected every 15 minutes. We explain this data set in more detail in Section 5.1. This

data set provides a vast quantity of inputs that far surpasses the amount of information used in the previous commercial and residential building studies. For example, the Great Energy Prediction Shootout data set contains only 5 measurements per hour. This means we are able to test existing techniques that were proven on previous smaller data sets, and introduce new techniques that have not previously been applied to this field.

4. Approach

We have tested seven different machine learning techniques on our residential data sets, and on the ASHRAE Building Energy Predictor Shootout data set. In this section, we briefly outline the technical details for each individual learner. In addition, we discuss advantages, disadvantages, and technical benefits for each technique. We present the techniques in the following order: Linear Regression; FFNN; SVR; Least Squares Support Vector Machines (LS-SVM); HME with Linear Regression Experts; HME with FFNN Experts; and Fuzzy C-Means with FFNN. Note, in the following subsections Y refers to the entire set of electrical consumption measurements, y refers to a single consumption measurement, X refers to the entire set of sensor observations, x_i refers to an individual sensor observation, and $\vec{x}$ refers to a vector of sensor observations.

4.1. Linear Regression

Linear Regression is the simplest technique, and can provide a baseline performance measure. Linear Regression is based on fitting a linear function with the following form:

$$y = \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \beta_n$$

Here, y is the target value, $x_1, x_2, ..., x_n$ are the available inputs, and β represents the functional weights. While this model is simplistic, it is used to establish a baseline performance for predicting electrical consumption on our residential data sets. If a technique performs worse than the baseline predictor, then it is most likely not appropriate for the data set.

4.2. Feed Forward Neural Network

As mentioned previously, previous studies have shown that Feed Forward Neural Networks (FFNN) are very capable at predicting electrical consumption. These previous studies have leveraged the fact that a FFNN can be

used as a general purpose method for approximating non-linear functions. That is, FFNN can learn to approximate a function f that maps $\Re^m \rightarrow \Re$ without making assumptions about the relationship between the input and outputs.

While a FFNN does not make assumptions about the inputs or outputs, it does require the user to define the model's structure, including the number of hidden layers and hidden units within the network and any other associated parameters. In this work, we explore a FFNN with a single hidden layer, which is the same overall structure as the previous studies. A FFNN with a single hidden layer for function approximation has the following mathematical representation:

$$f(x) = \sum_{j=1}^{N} w_j \Psi_j \left[\sum_{i=1}^{M} w_{ij} x_i + w_{io} \right] + w_{jo}$$

where N represents the total number of hidden units, M represents the total number of inputs, and Ψ represents the activation function for each hidden unit. In this work we selected $\tanh(x)$ as our activation function because prior research has shown good performance using this function (Dodier and Henze, 2004; Yang et al., 2005; Gonzalez and Zamarreno, 2005; Karatasou et al., 2006).

A FFNN's weights are learned using gradient descent-based methods, such as Newton-Raphson, by minimizing a user-specified error function. There are many possible error functions, such as Mean Squared Error (MSE), Sum Squared Error (SSE), and Root Mean Squared Error (RMSE). In this work, we use the SSE function.

However, a gradient descent learning approach poses two problems. The first problem is over-fitting. The FFNN can adjust its weights in such a way that it performs well on the training examples, but it will be unable to produce accurate responses for novel input examples. This problem is addressed by splitting the training set into two parts – a set used for training and a set for validation. When the error increases on the validation set, the learning algorithm should halt, because any further weight updates will only result in over-fitting the training examples.

The second problem involves avoiding local minima and exploring the search space to find a globally optimal solution. A local minimum is a point at which it is impossible to further minimize the objective function by following the gradient, even though the global minimum is not reached. However, it is

not possible to determine if any particular set of weights is a globally optimal solution or a local minimum. It is not possible to completely address this problem, but it is possible to avoid shallow local minima by using momentum and an adaptive learning rate. Momentum incorporates a small portion from the previous weight changes into the current weight updates. This can allow the FFNN to converge faster and to possibly step over shallow local minima. An adaptive learning rate dynamically changes the gradient descent step size, such that the step size is larger when the gradient is steep and smaller when the gradient is flat. This mechanism will allow the learning algorithm to escape local minima if it is shallow enough.

4.3. Support Vector Regression

Support Vector Regression (SVR) was designed and developed to minimize structural risk (Smola, A.J. and Schólkopf, B., 2004). That is, the objective is to minimize the probability that the model built from the training examples will make errors on new examples by finding a solution that best generalizes the training examples. The best solution is found by minimizing the following convex criterion function:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i + \xi_i^*$$

with the following constraints:

$$y_i - w^T\varphi(\vec{x_i}) - b \leq \epsilon + \xi_i$$
$$w^T\varphi(\vec{x_i}) + b - y_i \leq \epsilon + \xi_i^*$$

In the above equations, ϵ defines the desired error range for all points. The variables ξ_i and ξ_i^* are slack variables that guarantee that a solution exists for all ϵ. C is a penalty term used to balance between data fitting and smoothness. Lastly, w are the weights for the regression, and φ represents a kernel function for mapping the input space to a higher dimensional feature space.

There is one major advantage within the SVR optimization formulation: there is a unique solution which minimizes a convex function. However, the unique solution is dependent upon providing C, ϵ, and the necessary parameters for the user-selected kernel function φ. There are many techniques for selecting the appropriate parameters, such as grid search with cross-validation,

leave-one-out cross-validation, and many more. The work of Smola, A.J. and Schólkopf, B. (2004) provides a detailed overview of the different tuning techniques. In this work, all parameter settings were determined via grid search with cross-validation using LIBSVM's provided utilities (Chang and Lin, 2011).

However, SVR does have a potential disadvantage: scalability. The convex criterion function is optimized using quadratic programming optimization algorithms. There are many different algorithms and each has its own advantages and disadvantages (Smola, A.J. and Schólkopf, B., 2004), but the primary disadvantages are generally memory requirements and speed. However, the data sets used in our work are not large enough for these issues to be a real concern.

4.4. Least Squares Support Vector Machine

Least Squares Support Vector Machine (LS-SVM) is very similar to SVR, but with two notable differences. The first difference is the criterion function, which is based on least squares. The second difference is that the problem constraints are changed from inequality to equality. These differences allow the optimization function to be formulated as:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i^2$$

with the following constraint:

$$w^T\varphi(\vec{x_i}) + b + \xi_i = y_i$$

One advantage LS-SVM has over SVR is that this modified criterion function does not require quadratic programming to solve the optimization problem. This allows LS-SVM to find solutions much faster by solving a set of linear equations. The set of linear equations and their solution are well documented in Suykens et al. (2002a). However, LS-SVM uses all data points to define its solution, while SVR only uses a subset of the training examples to define its solution. This means that LS-SVM loses the sparsity property, which may or may not affect the solutions' ability to generalize. However, there are studies that address the sparsity issue through pruning or via weighting the examples (Suykens et al., 2002b; Hoegaerts et al., 2004).

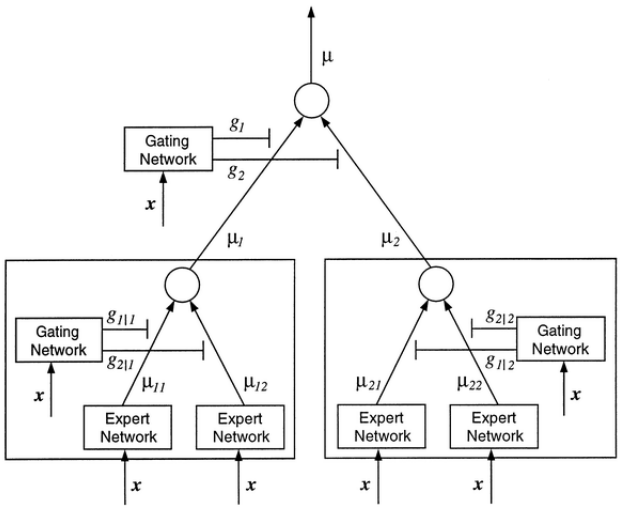

Figure 1: An example Hierarchical Mixture of Experts model with depth 2 and branching factor 2. This figure is provided by Jordan and Jacobs (1994).

4.5. Hierarchical Mixture of Experts

Hierarchical Mixture of Experts is a type of Neural Network that learns to partition an input space across a set of experts, where the input space in our application is the raw sensor values. These experts will either specialize over a particular region, or assist each other in learning a region or regions. These HME models are very useful for exploring the possibility that a data set contains multiple regimes or sub-populations. For example, a residential home's electrical consumption can vary according to the seasons – fall, winter, spring, and summer. These variations may be best explained by separate individual models. An HME model tries to discover these different sub-models automatically, and fit an Expert to each sub-model. While the previous motivating example implies temporal based sub-model changes, the HME model can only detect sub-model changes by using spatial differences, as well as using each expert's ability to produce accurate predictions during training.

HME models are constructed using two types of networks: Gating and Expert networks. Figure 1 presents an example HME with two layers of Gating networks and four Expert networks. This particular HME is modeled

as:

$$\mu = \sum_i g_i \sum_{j|i} g_{j|i} F_{ji}(\vec{x})$$

where g_i represents the top level gating network's output, $g_{j|i}$ represents the outputs from the lower level gating networks, and $F_{ji}(\vec{x})$ represents the output from an Expert network. This example model is easily extended to have additional Gating networks and Experts by adding additional summations.

The Gating network probabilistically partitions the input space across either additional Gating or Expert networks. The partitioning is achieved using the following softmax function:

$$g_i = \frac{e^{\xi_i}}{\sum_{k=1}^{N} e^{\xi_k}}$$

where ξ represents the Gating network outputs, g_i is the normalized weight associated with the ith sub-network, and N represents the total number of sub-networks. Each Gating network approximates the conditional probability $P(Z|X)$ in which Z represents the set of direct sub-networks and X represents the set of observations. Approximating $P(Z|X)$ allows the Gating network to determine which Expert network or networks is more likely to produce an accurate prediction.

Each Expert network represents a complete learning system. However, unlike a standalone learning system, each Expert is expected to specialize over different regions defined by the Gating networks. In the original HME studies, the only supported expert learner was Neural Networks (Jordan and Jacobs, 1992). However, a later extension on the work introduced support for Linear Regression Experts (Jordan and Jacobs, 1994). While these studies only presented Neural Network and Linear Regression Experts, the learning procedures introduced in the extension do not limit the Experts to only these learning systems. The only restriction placed on the Experts is that they have an associated likelihood function. For example, the assumed likelihood function in these previous studies for regression problems is that each Expert's error rate follows a Gaussian distribution.

The original studies present three different maximum likelihood learning algorithms. The first algorithm is based on using gradient ascent. Using the HME shown in Figure 1 as an example, all three algorithms attempt to

maximize the following likelihood function:

$$L(Y|X,\theta) = \prod_t \sum_i g_i^{(t)} \sum_j g_{j|i}^{(t)} P_{ij}(y^{(t)}|\vec{x}^{(t)}, \theta_{ij})$$

where P_{ij} represents an Expert's likelihood function and θ represents parameters associated with each Gating network and with each Expert.

The other two algorithms approach the problem as a maximum likelihood problem with missing data. The missing or unobservable data is a set of indicator variables that specify the direction for partitioning the input space at each Gating network. If all indicator variables are known, then maximizing the HME's likelihood function is split into two separate problems (Jordan and Jacobs, 1994). The first problem is learning the parameters for each individual Gating network, while the second problem is training each Expert on the appropriate training examples. Given that it is generally impossible to know the exact value for each indicator variable in advance, the original developers derived two different Expectation Maximization (EM) (Dempster et al., 1977) algorithms. The first algorithm is an exact EM algorithm and the second algorithm approximates the first algorithm.

In addition to FFNN and Linear Regression Experts, we extended the Mixture of Experts (MoE) with LS-SVM Experts, by Lima et al. (2009), to Hierarchical Mixtures. The Maximization process is presented as a weighted regression problem in both HME EM algorithms, which implies any learning system that supports weighted examples can be used as an Expert. We utilize this property and the robust LS-SVM work by Suykens et al. (2002b) to integrate LS-SVM Experts into the HME framework. However, we found that the results for HME with LS-SVM on our residential data set and the Great Energy Prediction Shootout data set were not statistically different from a single LS-SVM. We believe this is due to all LS-SVM Experts using the same parameter settings as the single LS-SVM model. The findings in Lima et al. (2009) suggest that the parameter settings can be the same across the LS-SVM Experts, but the parameter settings should be determined by searching the parameter space using the entire mixture model.

4.6. Fuzzy C-Means with Feed Forward Neural Networks

An alternative approach to HME is to separate the learning process into two steps. The first step is an unsupervised learning phase that uses clustering to approximate $P(Z|X)$, and the second step is to use each cluster

to train the Experts. It is possible to use any clustering algorithm, such as K-Means, Self-Organizing Maps, Hierarchical Clustering, etc. However, a clustering algorithm that does not allow observations to belong to multiple clusters will produce very rigid approximations. A rigid approximation will cause Experts to ignore large sets of observations, which can cause the Experts to produce very poor models. This means each Expert will be less likely to produce reasonable responses when accounting for errors in the approximated $P(Z|X)$. We avoid rigid approximations by using Fuzzy C-Means (FCM), which allows for observations to belong to multiple clusters.

FCM is based on minimizing the following criterion function:

$$\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\|\vec{x}_i - \vec{c}_j\|^2$$

where u_{ij} represents the probability that $\vec{x}_i$ is a member of cluster $\vec{c}_j$, and m is a user-defined parameter that controls how much an observation can belong to multiple clusters. The criterion function is minimized through an iterative process using the following equations:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m}\vec{x}_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \frac{\|\vec{x_i}-\vec{c_j}\|}{\|\vec{x_i}-\vec{c_k}\|}^{\frac{2}{m-1}}}$$

Iterating over the above equations will produce N cluster centroids and a weight matrix U. N represents the total number of user-defined clusters and each row in U represents an instance of $P(Z|X)$. The weight matrix can be used to train a Gating network or for weighting the training examples when fitting the Experts. In this work, we choose to use the second option, and use N cluster centers to approximate $P(Z|X)$ for new observations by computing the second equation.

While we implemented FFNN, Linear Regression, and LS-SVM Experts for the HME models, we have only explored FFNN Experts for this two-step approach. This approach is not limited to FFNN Experts, and can support all learning systems that can incorporate weighted training examples. In addition, the likelihood function requirement for the Experts is removed.

While this approach seems superior to the HME, it relies on the critical assumption that the spatial relation between observations can approximate $P(Z|X)$, while HME approximates $P(Z|X)$ by maximizing $P(Y|X,\theta)$.

4.7. Temporal Dependencies

In the realm of function approximation, *temporal dependencies* mean that the target response y_t is dependent on past observations, $\vec{x}_{t-1}$, as well as current observations $\vec{x}_t$. These temporal dependencies either follow a Markov order or are sparse. If the dependencies follow a Markov order, then the response y_t is dependent on previous complete sets of observations. For example, if y_t has temporal dependencies with Markov order 2, then it is dependent on $\vec{x}_t$, $\vec{x}_{t-1}$, and $\vec{x}_{t-2}$. However, sparse temporal dependencies indicate that y_t can be dependent on any combination of past observations rather than a complete set. Exploring all possible sparse temporal dependencies grows exponentially and is thus intractable.

Our work focuses on predicting future hourly electrical consumption. This means we can only use observations $\vec{x}_{t-1}$, $\vec{x}_{t-2}$, etc., to predict y_t. If we did not follow this constraint, we would use future information to predict y_t. Therefore, Markov order 1 models use observation $\vec{x}_{t-1}$, order 2 models use observations $\vec{x}_{t-1}$ and $\vec{x}_{t-2}$, and so forth.

In previous studies, researchers explored sparse temporal dependencies either with manual statistical testing or automatically, by defining a feasible search space within the learning system. The winner for the first Shootout, which we discussed previously, used ARD to automatically determine the relevant inputs. The possible inputs included temporal dependencies. However, the total number of available inputs for the competition was fairly small. For example, the winner's FFNN used 25 different inputs, while a single order 3 model uses approximately 432 inputs. Therefore, we only consider the entire set of inputs, rather than trying to search for the best inputs. We are currently exploring scalable automatic methods that can help identify the sparse temporal dependencies, yet these methods present considerable research challenges and are beyond the scope of this article.

4.8. Model Selection

Each presented learning system has a variety of different parameters. Some parameters are estimated during the learning process, while others are user-defined parameters. Each different combination of learned parameters and user-defined parameters constitutes a single model configuration.

In order to determine which learning system performs best at predicting residential electrical consumption, we need to select the best model configurations for each technique and compare these best configurations. This type of comparison facilitates a fair comparison across all techniques.

There are several different model selection techniques. For example, cross-validation methods help find parameter estimates that can generalize to unseen data by periodically testing the current model on a validation set. Another cross-validation method, called K-Folds cross-validation, ensures that each data point is used as a testing example at least once, and that the training and testing sets are fixed. This means that each learning system can be compared using the same testing and validation sets, which is ideal for determining how different user-defined parameters affect the models.

We use a combination of cross-validation and K-Folds cross-validation to select the best model for each technique. We separate out a cross-validation set from the allocated training data, which leaves each learning system with a training set, a validation set, and a testing set. However, the Linear Regression models do not utilize the validation set, because the parameters are estimated using a non-iterative maximum likelihood method. We then select the model from each technique that has the best performance across all the testing sets. This allows us to identify models that generalize well to unseen data, and determine which user-defined parameters settings are best for each learning system.

5. Methods

5.1. Campbell Creek Data

The new residential data set used in our research, called the Campbell Creek data set, is a rich and unique data set. This data set was collected from three different homes located in west Knox County, Tennessee. These Campbell Creek homes are leased and operated by Tennessee Valley Authority (TVA) as part of a study testing energy efficient materials and their savings (Christian et al., 2010). The first house in this study, called House 1, is a standard two-story residential home. However, the second and third house, called House 2 and House 3 respectively, were modified to decrease energy consumption. House 2 uses the same construction materials as House 1, but was retrofitted with more energy efficient appliances, water heater, and HVAC. House 3 was built using construction techniques and materials

designed to help reduce energy consumption. In addition, House 3 has a set of photovoltaics for generating electricity and solar thermal water heater.

In this dataset each house has approximately 140 different sensors that collect data every 15 minutes. Each house is also outfitted with automated controls that manage the opening/closing of the refrigerator door, using the oven, running clothes washer and dryer, as well as shower usage. These automatic controls achieve an occupancy pattern that is consistent with typical energy usage patterns of American households, as determined by a Building America study (Hendron et al., 2010). The simulated occupancy provides stable behavioral patterns across all three homes, making device usage within the data set consistent across test environments. This means the data set is free from behavioral factors, making it easier to compare results for different houses. Note that this data set provides a vast quantity of inputs that far surpasses the amount of information used in the previous commercial and residential building studies.

Removing the dynamic human behavior is clearly advantageous for making better predictions. However, these three homes were used to conduct numerous experiments throughout the data collection process. This means there were equipment substitutions, thermostat set point changes, prototype equipment test, and much more. Therefore, the data sets still exhibits rich dynamic behavior, unless the data collected from these experiments is removed or treated as special cases.

5.2. Experimental Design

Our primary interest is determining which models perform the best at predicting electrical consumption for the next hour. We facilitate this process by testing each technique under a number of different configurations and by a combination of K-Folds and Cross Validation. Each model is trained and tested using 10-Folds, which were created from sensor data collected in each Campbell Creek House from January 1, 2010 until December 31, 2010. In addition, if a model supports Cross Validation, such as a FFNN, its training set is split into a training set and validation set. The split settings are the same for all models – 85% for training and 15% for validation. Only SVR and Linear Regression do not make use of a validation set during training. However, SVR uses a validation set when tuning the model's external parameters.

The different model configurations include testing: Markov order 1 through 3, different numbers of hidden neurons, different numbers of clusters, and dif-

ferent complete tree structures. Every model that incorporates a FFNN was tested with 10 to 15 hidden neurons. The Fuzzy Cluster approach was tested with two to eight clusters. Lastly, for all HME models, we tested complete trees with depths 1 through 3 and branching factors 2 through 4. Testing these different settings has allowed us to select the best model configuration for each technique and facilitates comparisons between different techniques.

In addition, we tested all techniques on the Great Energy Predictor Shootout. These experiments use two types of sensor inputs. The first, called S1, includes only environmental sensors and time information, while the second, called S2, includes environmental sensors, time information, and actual previous electrical consumption. The sensor inputs and naming conventions follow those presented in (Karatasou et al., 2006; Li et al., 2011). In this work, S1's inputs are defined as follows:

$$\text{S1} : \vec{x}(t) = (T(t), S(t), s, sh, ch)$$

where $T(t)$ is the current temperature, $S(t)$ is the current solar flux, s is an indicator variable, sh is the sin of the current hour, and ch is the cos of the current hour. The indicator variable s, is used to denote whether the current day is a holiday or weekend. The variable is set to 1 for all holidays and weekends, and set to zero for all workdays. S2's inputs are defined as follows:

$$\text{S2} : \vec{x}(t) = (y(t-1), y(t-2), T(t), S(t), s, sh, ch)$$

where $y(t-1)$ and $y(t-2)$ represent known electrical consumption values for the previous two hours.

5.3. Performance Metrics

The primary measure for selecting the winners in the ASHRAE competition was the Coefficient of Variance (CV) measure (Kreider and Haberl, 1994), which determines how much the overall prediction error varies with respect to the target's mean. A high CV score indicates that a model has a high error range. The CV measure is defined as follows:

$$\text{CV} = \frac{\frac{1}{N-1}\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\bar{y}} \times 100$$

where $\hat{y}_i$ is the predicted energy consumption, y_i is the actual energy consumption, and $\bar{y}$ is the average energy consumption.

A second metric, Mean Bias Error (MBE), was used to break ties within the competition. This metric establishes how likely a particular model is to over-estimate or under-estimate the actual energy consumption. A MBE closest to zero is preferred, because this means the model does not favor a particular trend in its prediction. The MBE measure is defined as follows:

$$\text{MBE} = \frac{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)}{\bar{y}} \times 100$$

where $\hat{y}_i$, y_i, and $\bar{y}$ represent the same components presented in the CV measure.

Another metric that is commonly used in the literature to assess regression accuracy is Mean Absolute Percentage of Error (MAPE) (Karatasou et al., 2006; Gonzalez and Zamarreno, 2005). The MAPE measure determines the percentage of error per prediction, and is defined as follows:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \times 100$$

where $\hat{y}_i$ and y_i represent the same components defined in the CV and MBE measures.

In this work, we use CV as our primary metric. MBE is the first tie breaker, and MAPE is the final tie breaker. We only take the tie breaker metrics into consideration when the CV metric does not measure a statistical difference between two techniques. If both original ASHRAE metrics are inconclusive, our decisions are based on the MAPE metric.

6. Results

Our experimental results are organized in the following order: ASHRAE Shootout 1, Campbell Creek House 1, Campbell Creek House 2, and Campbell Creek House 3. Each subsection presents the best performing models from the 7 techniques. Following these result sections, we present a results summary, which presents the best general overall technique and highlights the key results for each data set.

6.1. Great Energy Prediction Shootout

For comparison purposes, we ran our 7 implemented machine learning techniques on the earlier Great Energy Prediction Shootout data set. The

S1

	CV(%)	MBE(%)	MAPE(%)
Regression	14.12±0.00	7.69±0.00	13.41±0.00
FFNN	**11.29±0.00**	**8.32±0.00**	**9.14±0.00**
SVR	11.93±0.00	8.95±0.00	9.63±0.00
LS-SVM	13.70±0.00	10.32±0.00	11.21±0.00
HME-REG	14.11±0.00	7.66±0.00	13.40±0.00
HME-FFNN	11.49±0.00	2.91±0.00	9.73±0.00
FCM-FFNN	11.51±0.00	8.71±0.00	9.45±0.00

S2

	CV(%)	MBE(%)	MAPE(%)
Regression	4.07±0.00	1.01±0.00	2.86±0.00
FFNN	2.93±0.00	0.64±0.00	1.77±0.00
SVR	3.97±0.00	1.41±0.00	2.31±0.00
LS-SVM	6.35±0.00	1.53±0.00	4.50±0.00
HME-REG	4.05±0.00	0.99±0.00	2.85±0.00
HME-FFNN	**2.75±0.00**	**0.52±0.00**	**1.60±0.00**
FCM-FFNN	**2.71±0.00**	**0.55±0.00**	**1.61±0.00**

Table 1: Great Energy Prediction Shootout Results. Best results are shown in bold font.

results for these experiments are presented in Table 1. We are not able to make statistical claims about the difference between techniques, because the original competition presented only a single training and testing set. However, the S1 results indicate that a FFNN is the best predictor for electrical consumption. This finding is consistent with the existing literature (Kreider and Haberl, 1994). However, all methods except Linear Regression, HME with Linear Regression, and LS-SVM are competitive with the best three competition winners: CV – 10.36%, 11.78%, 12.79%.

The S2 results in Table 1 suggest that HME with FFNN and FCM with FFNN are better than the FFNN. The existing published results for the S2 inputs range from 2.44% to 3.65% (Karatasou et al., 2006; Li et al., 2011). From these results, we can conclude that Neural Network type methods perform best on this data set. We can also conclude that LS-SVM is the worst advanced technique, with Linear Regression and HME with Linear Regression being only slightly better.

6.2. Campbell Creek House 1

Table 2 presents the results from applying all the techniques to House 1 with different Markov orders. These results illustrate which techniques perform the best on House 1 and the effects that different Markov orders have on these techniques. Almost all techniques increase in performance as the order increases. The three methods that do not increase in performance are FFNN, HME with FFNNs, and FCM with FFNNs. The FFNN results are not statistically different across all orders. The other two techniques show performance increases with order 2, but order 3 is not statistically different.

According to the CV metric, the best techniques are the order 2 SVR, order 2 LS-SVM, order 2 HME with FFNNs, and order 2 FCM with FFNNs. While the CV performance for the SVR model is not significantly different, its MBE error is statistically different from the other techniques, illustrating

House 1

Order 1

	CV(%)	MBE(%)	MAPE(%)
Regression	32.38±1.91	-0.06±1.08	30.52±1.41
FFNN	25.10±2.34	0.66±1.43	21.08±1.14
SVR	24.60±1.78	-2.46±0.95	17.05±0.94
LS-SVM	23.39±1.26	0.01±0.84	18.21±0.89
HME-REG	32.35±1.82	-0.05±1.02	30.57±1.42
HME-FFNN	22.77±1.56	0.15±0.98	17.74±0.65
FCM-FFNN	22.65±1.42	0.81±0.95	18.18±0.75

Order 2

	CV(%)	MBE(%)	MAPE(%)
Regression	27.63±1.95	-0.03±1.09	26.18±1.51
FFNN	24.32±2.61	0.53±1.74	22.28±2.67
SVR	21.58±1.40	-1.41±0.89	16.41±0.95
LS-SVM	**20.05±0.81**	**0.06±0.62**	**16.11±0.85**
HME-REG	27.60±2.13	-0.03±1.01	26.11±1.67
HME-FFNN	20.15±1.65	0.46±0.93	17.07±1.19
FCM-FFNN	20.53±1.76	0.74±0.87	17.57±1.42

Order 3

	CV(%)	MBE(%)	MAPE(%)
Regression	26.27±1.19	-0.11±1.45	24.33±0.96
FFNN	25.24±1.59	1.00±1.05	22.29±1.81
SVR	21.32±1.32	-1.50±0.80	15.48±0.87
LS-SVM	20.36±1.46	0.11±0.63	15.73±1.11
HME-REG	26.14±1.10	-0.08±1.44	24.21±0.93
HME-FFNN	20.39±1.67	0.70±0.92	17.09±0.81
FCM-FFNN	21.03±1.29	0.47±1.49	18.27±1.06

Table 2: Results for all techniques applied to Campbell Creek House 1. Best results are shown in bold font.

that it has potential to perform much poorer than the other three techniques. In addition, the other three techniques do not have significantly different MBE results. Even though the second tie-breaker metric does not indicate a single best model, the third tie-breaker (MAPE) shows clearly that LS-SVM has the best MAPE measure and is statistically different from HME with FFNNs and FCM with FFNNs. Therefore, LS-SVM is the best model for predicting next hour energy consumption for House 1.

6.3. Campbell Creek House 2

The results for House 2 (Table 3) show a different performance trend as the Markov order increases, compared to House 1. While most techniques illustrated an increase in performance on House 1 as the order increased, these techniques only present small improvements on House 2. The improvements are only statistically significant for the baseline Linear Regression technique and order 3 SVR.

Given the minimal performance gains from the increasing orders and the CV results for House 2, the best techniques are order 1 LS-SVM and Order 1 FCM with FFNNs. The order 1 models are selected over the Order 2 and 3 models, because the three models are not statistically different within an acceptable confidence, and higher order models are much more complex. The higher order models are more complex because as the number of inputs

House 2

Order 1

	CV(%)	MBE(%)	MAPE(%)
Regression	36.73±2.26	-0.13±1.00	31.01±3.48
FFNN	33.24±1.26	0.50±0.91	27.28±3.12
SVR	30.36±1.83	-2.95±1.03	20.44±2.81
LS-SVM	**27.88±1.24**	**-0.05±0.91**	**20.47±2.37**
HME-REG	35.82±1.04	0.15±0.88	30.48±3.20
HME-FFNN	29.30±1.28	0.09±1.01	22.71±2.92
FCM-FFNN	**28.14±1.21**	**0.40±0.97**	**21.96±2.74**

Order 2

	CV(%)	MBE(%)	MAPE(%)
Regression	34.15±1.66	0.05±1.61	28.36±3.72
FFNN	33.83±1.98	0.21±1.45	27.07±4.14
SVR	29.22±1.06	-3.00±1.12	19.42±3.27
LS-SVM	27.43±1.90	0.20±1.03	20.17±2.26
HME-REG	34.15±1.74	0.14±1.38	28.29±3.86
HME-FFNN	28.17±2.04	0.26±0.58	22.43±2.44
FCM-FFNN	28.34±1.67	-0.20±1.27	22.30±3.28

Order 3

	CV(%)	MBE(%)	MAPE(%)
Regression	33.15±1.33	-0.02±0.96	27.87±2.40
FFNN	34.23±1.63	2.01±2.45	29.62±2.16
SVR	28.59±2.05	-2.33±1.09	19.58±2.07
LS-SVM	27.68±1.91	-0.02±1.71	20.23±2.56
HME-REG	33.20±1.32	-0.08±0.97	27.95±2.31
HME-FFNN	29.64±2.21	-0.12±1.64	24.81±0.38
FCM-FFNN	28.94±1.46	0.45±1.27	22.76±2.03

Table 3: Results for all techniques applied to Campbell Creek House 2. Best results are show in bold font.

increases, the total number of parameters to estimate increases. A more complex model has less potential to generalize to new examples, which makes it less desirable when simpler models provide equal performance. In addition, the tie breaker measures MBE and MAPE are not statistically different for all orders.

6.4. Campbell Creek House 3

The results for House 3, shown in Table 4, present the same trend as the House 2 results. As the order increases, most techniques have minimal or no performance gains. The only models that present statistically significant improvements are order 3 SVR and order 2 LS-SVM. The order 3 SVR shows improvement in the CV measure, while the order 2 LS-SVM presents improvement in the MAPE measure. All other models are not statistically different within a reasonable confidence range across the different orders.

According to the results in Table 4, order 3 SVR's CV value is statistically different from every model except order 2 and 3 LS-SVMs' CV values. In addition, order 1 LS-SVM's CV value is not statistically different from all HME with FFNN models and FCM with FFNN models, but the CV values for orders 2 and 3 are statistically better. Therefore, order 2 LS-SVM and order 3 SVR are the best models based on the CV measure. The order 3

House 3

Order 1

	CV(%)	MBE(%)	MAPE(%)
Regression	40.07±2.21	0.07±1.15	32.49±1.88
FFNN	37.15±1.57	0.35±2.03	28.92±2.55
SVR	33.71±1.72	-3.36±0.99	21.49±1.80
LS-SVM	31.60±2.07	-0.15±1.10	22.25±1.33
HME-REG	39.17±2.17	0.33±1.38	31.72±2.07
HME-FFNN	32.98±1.28	-0.12±0.84	23.99±1.63
FCM-FFNN	33.03±1.67	0.93±1.52	25.28±2.14

Order 2

	CV(%)	MBE(%)	MAPE(%)
Regression	39.26±4.19	0.11±1.86	31.34±2.58
FFNN	38.02±2.49	2.05±2.67	29.83±2.02
SVR	32.38±2.96	-3.12±1.73	20.72±1.38
LS-SVM	**30.66±2.53**	**-0.05±0.93**	**21.33±1.40**
HME-REG	38.48±4.34	1.03±1.72	30.53±3.07
HME-FFNN	32.99±2.17	1.07±1.17	24.76±1.94
FCM-FFNN	32.92±2.49	0.76±2.03	24.20±2.06

Order 3

	CV(%)	MBE(%)	MAPE(%)
Regression	38.53±3.47	0.15±1.22	30.49±2.15
FFNN	38.58±2.07	-0.08±2.46	30.57±2.51
SVR	31.88±2.01	-2.84±0.97	20.47±1.69
LS-SVM	30.78±2.56	-0.21±1.04	21.36±1.50
HME-REG	38.22±3.58	1.20±1.49	29.52±2.47
HME-FFNN	33.34±1.83	1.09±1.24	25.15±2.13
FCM-FFNN	33.66±2.09	1.17±1.30	25.51±1.72

Table 4: Results for all techniques applied to Campbell Creek House 3. Best results are shown in bold font.

LS-SVM model is excluded because it is not statistically different from the simpler order 2 model.

Note that the House 3 results indicate that SVR demonstrates a large MBE measure for all Markov orders. This means that the SVR model is removed from consideration based on the second tie-breaker measure. Therefore, the best technique for predicting next hour energy consumption for House 3 is LS-SVMs.

6.5. Results Summary

Our findings indicate that FFNN performs best on the original ASHRAE Shootout data set, which is consistent with the literature. However, our results for S2 indicate that other Neural Network methods might perform better. This is consistent with the recent work in (Li et al., 2011).

Our findings also indicate that traditional methods, such as FFNN, are not the best overall method for predicting future residential electrical consumption. In fact, on House 3 the FFNN's performance is extremely close to the baseline performance established by Linear Regression. Traditional methods perform better on House 1 and 2, but not as well as other techniques.

Despite traditional methods not performing as well on the residential data sets, our results establish that FCM with FFNN, HME with FFNN, and LS-

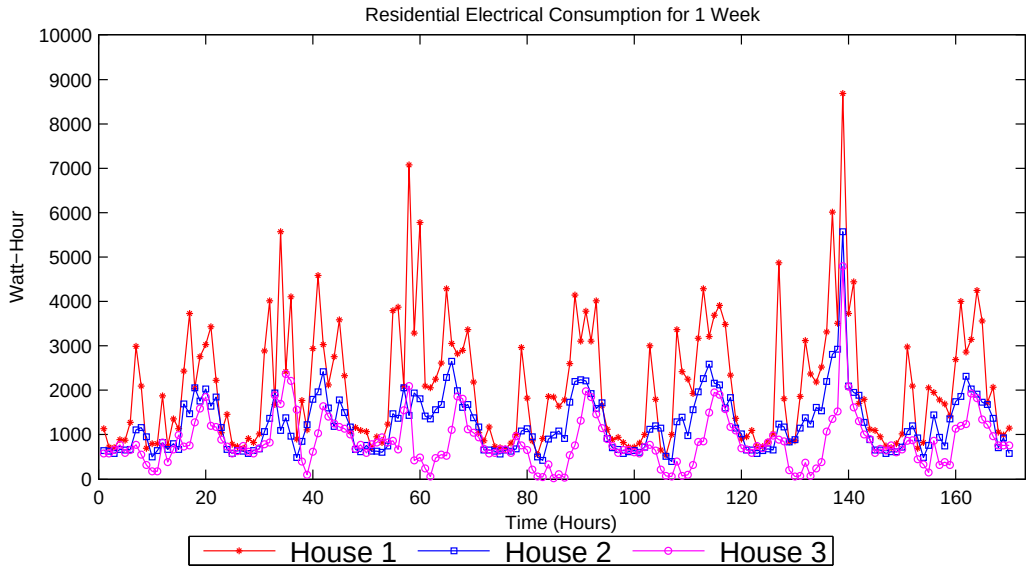

Figure 2: This figure presents one week of electrical consumption for all three residential homes, from the second week in September, 2010.

SVM work well on all three houses. However, LS-SVM is statistically the best technique at predicting future residential electrical consumption over the next hour.

7. Discussion

The different performance results for each house stem from the fact that each house is fundamentally different. These physical differences make each house have a very different energy response pattern, even though each house is automated to run exactly the same schedule. Figure 2 illustrates the electrical consumption for a single week in September. The complexity of the energy patterns exhibited by Houses 2 and 3 make them harder to predict than House 1. The figure shows that House 3 is prone to sudden drops in electrical consumption, while House 2's electrical consumption fluctuates much more frequently. House 1 may appear to fluctuate as sharply as House 2, but the fluctuations are much less on average. The physical differences certainly impact the physical sensor data as well.

The results from the Great Energy Predictor Shootout and results from

predicting electrical consumption in other commercial buildings have established expected ranges for good CV values – on the order of 2% to 13%, according to the existing literature. The results are clearly dependent on the input variables, but a learning approach is generally considered acceptable if it is within that range. However, we note that our residential results are not within this range. These results are not due to the learning approaches being implemented incorrectly or poorly. In fact, all learning approaches are implemented using existing or modified software packages. The LS-SVM implementation is from LS-SVMlab (Suykens et al., 2002a), the SVR implementation is from LIBSVM (Chang and Lin, 2011), the HME implementation uses modified software provided by the authors of (Martin et al., 2004), and all remaining learning systems are implemented using existing MATLAB modules provided by Mathworks. Considering the reasonable performance of these same techniques on the Great Energy Prediction Shootout data set (Table 1) and the fact that all techniques are built using established software, the only possible cause for not matching the established CV range is that each house has more complex energy usage patterns than typical commercial buildings.

Comparing the residential electrical consumption (Figure 2) with the commercial electrical consumption (Figure 3) shows that commercial buildings have fairly stable usage patterns and less sudden change than residential buildings. The reason for this difference is based purely on the size of the buildings, and the fact that small variations in consumption do not significantly affect the overall consumption. A larger building will obviously consume more electricity and contain more people, which means that the actions of a few individuals turning on lights or using additional electricity will have very little effect on the buildings' consumption trend. However, in a smaller building, minor changes to the environment can cause noticeable effects. For example, turning all the lights on in most houses will cause more noticeable fluctuation than turning on the equivalent number of lights in a commercial building.

In addition, residential buildings exhibit more complex usage patterns. Figure 4 illustrates three weeks of measured electrical consumption for House 3. The usage patterns are very similar for the first two weeks and share similar highs and minimums. However, the usage pattern completely changes during the third week (hours 315 through 500). This variability is mostly dependent upon the house's ability to produce solar power and how much solar power the house is able to produce. While this figure illustrates changes

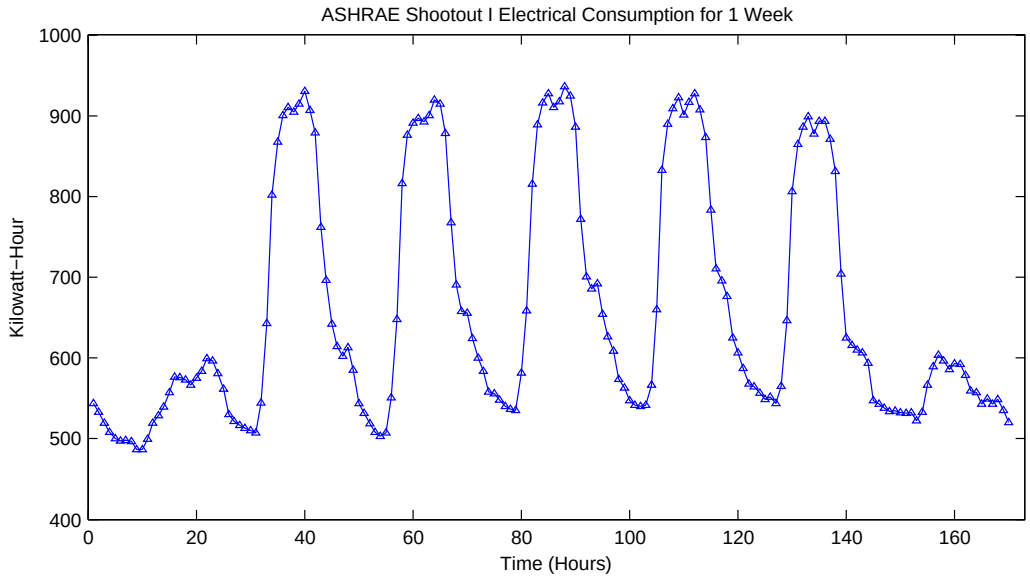

Figure 3: One week of electrical consumption for the Great Energy Prediction Shootout building, from the second week in September, 1989.

in consumption patterns for House 3, changes in consumption patterns are not unique to House 3 and also occur in Houses 1 and 2; the pattern changes are just more pronounced in House 3.

The Great Energy Prediction Shootout data set does contain changes in consumption patterns, but these changes correspond with holidays, weekends, and normal vacation periods. On the other hand, the changes in these residential homes is dependent on environmental variables and changes in occupant behavior. Thus, these three homes provide a rich and interesting data set for modeling energy prediction that is more challenging than the currently available commercial data sets.

According to the results presented in Tables 2, 3, and 4, changing the Markov order had varying affects. Most techniques applied to House 1 showed a statistically significant performance increase as the order was increased from 1 to 2. On House 1, fewer techniques present improvement by increasing the order even further. However, most techniques applied to Houses 2 and 3 show very little or no performance gains as the order increases. On House 2 only Linear Regression shows statistically significant improvements by increasing the order. In addition, only two techniques show statistically

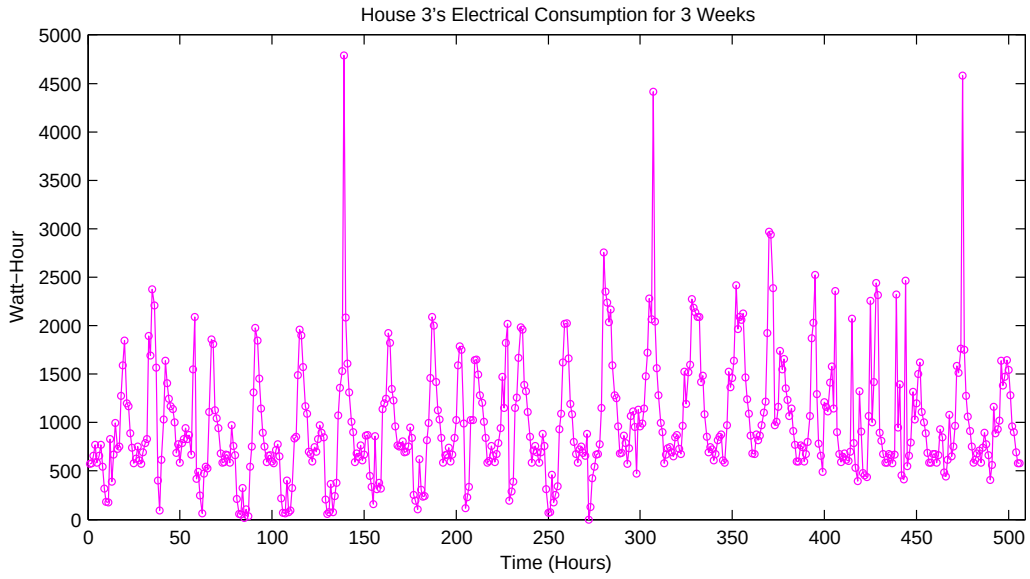

Figure 4: Three weeks of electrical consumption for House 3, starting from the second week in September, 2010.

significant improvement on House 3: LS-SVM and SVR.

There are two possible explanations for these results. First, the temporal dependencies could extend back much further in time than order 3. Second, the consumption patterns could change often enough that increasing the past observations does not help predict future consumption. The first option is possible, but requires further testing and evaluation. However, extending the order further without removing irrelevant inputs may cause most models to perform worse than the ones with smaller orders, due to overfitting. Therefore, this requires testing higher orders and determining the most relevant inputs for predicting electrical consumption. We are actively exploring methods for determining the most relevant inputs, but reporting these results is beyond the scope of this paper.

The second option is the most plausible explanation. Houses 2 and 3 change consumption patterns fairly often, and are dependent on future events that are not always represented within past observations. For example, House 3's ability to generate solar power is dependent on external weather events that are not guaranteed to follow a regular pattern. However, House 2 is more difficult to explain. House 2's consumption pattern changes regularly, except

ASHRAE Shootout

S1

	CV(%)	MBE(%)	MAPE(%)
Regression	13.26±0.16	-0.02±0.43	11.64±0.11
FFNN	**8.81±0.17**	**0.01±0.10**	**7.10±0.09**
SVR	9.16±0.23	0.05±0.04	7.48±0.12
LS-SVM	**8.85±0.18**	**0.02±0.21**	**6.95±0.21**
HME-REG	13.26±0.15	0.03±0.41	11.65±0.10
HME-FFNN	**8.74±0.22**	**-0.02±0.04**	**7.00±0.11**
FCM-FFNN	**8.74±0.26**	**0.05±0.24**	**6.99±0.21**

S2

	CV(%)	MBE(%)	MAPE(%)
Regression	4.01±0.35	0.00±0.27	2.71±0.08
FFNN	2.29±0.16	0.06±0.12	1.51±0.05
SVR	3.27±0.36	0.09±0.16	1.90±0.12
LS-SVM	3.77±0.44	-0.07±0.08	2.13±0.20
HME-REG	4.01±0.35	0.01±0.29	2.70±0.10
HME-FFNN	**2.20±0.19**	**-0.03±0.07**	**1.39±0.01**
FCM-FFNN	**2.17±0.17**	**0.01±0.11**	**1.38±0.00**

Table 5: Great Energy Prediction Shootout results using 3-Folds. The data set's order was randomized before being divided into folds. Each test set has approximately the same number of examples as the original competition test set. Best results are shown in bold font.

that there are periods where the electrical consumption sporadically increases more than the normal trends. These instantaneous changes in patterns are not represented by past observations, which means increasing the order will not necessarily help.

Our residential results establish that LS-SVM is the best technique from the ones we explored. However, the Shootout results establish that this technique only performs better than HME with Linear Regression and Linear Regression alone. Clearly the LS-SVM model fails to generalize to the Shootout testing data. The model failed to generalize because the provided training data is not general. The electrical response signal for the training data and testing data are statistically different, but LS-SVM uses every training example to help define its model. This means that the LS-SVM builds a model that expects the testing response to resemble the observed training reponse. However, in this situation the electrical consumption pattern changes and the LS-SVM model is not able to predict these changes. We were able to test this idea by randomizing the Shootout training and testing data, such that the sets were more similar.

Our experiments with this modified data set show a performance increase for most techniques (Table 5). More importantly, LS-SVM is now a more competitive learning algorithm on this data set when presented with a more general training set. In our residential experiments, we shuffled the data sets before dividing the data into folds. This allowed us to perform all experiments with training and testing data sets that covered a wide range of different scenarios. Ultimately, we plan to train all methods on the entire 2010 Campbell Creek data set and perform tests on the entire 2011 Campbell

Creek data set once the year is complete.

8. Conclusion and Future Work

Given sensor data collected from three residential homes, we aimed to determine which machine learning technique performed best at predicting whole building energy consumption for the next hour. Our results show that LS-SVM is the best technique for predicting each home's future electrical consumption. In addition, our results show that the previously accepted method, FFNNs, performs worse than the newer techniques explored in this work: HME-FFNN, LS-SVM, and FCM-FFNN. Lastly, our results show that SVR and LS-SVM perform almost equally with respect to CV and MAPE. However, experiments with SVR present poor MBE results, which makes LS-SVM the preferred technique.

In addition, we validated our methods by producing comparable results on the Great Energy Prediction Shootout data set. These validation results are consistent with the existing literature in concluding that FFNN performs best on the original competition data set, and that other types of Neural Networks might perform even better. In addition, our results show that the LS-SVM is the worst performing technique for the Shootout data set, and that shuffling the data improves its performance.

In future work, we will explore which sensors are most important for predicting residential electrical consumption. This is a pivotal problem that must be solved, because it is not practical to install and support 140 sensors for new homes. Extending the work by Bozdogan and Haughton (1998) will allow us to produce approximate solutions to this problem. Approximate solutions are required, because the total number of possible sensor combinations is combinatoric. This means that for a large enough number of sensors, it is not possible to explore all possible sensor combinations. Even approximate solutions will help mitigate cost and overhead, making sensor-based energy modeling more viable.

9. Acknowledgements

This work was funded by field work proposal CEBT105 under the Department of Energy Building Technology Activity Number BT0305000. The Campbell Creek research project was funded by the Tennessee Valley Authority (TVA). The authors would like to thank David Dinse (TVA CC

project manager), Jeff Christian (ORNL CC project manager), Tony Gehl (ORNL data acquisition system and entropy warrior), and Philip Boudreaux (occupancy simulation expert) for providing the data that made this study possible.

Oak Ridge National Laboratory is managed by UT-Battelle, LLC, for the U.S. Dept. of Energy under contract DE-AC05-00OR22725. This manuscript has been authored by UT-Battelle, LLC, under Contract Number DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 2009. 2009 ASHRAE Handbook - Fundamentals. D&R International, Ltd.

Bozdogan, H., Haughton, D., 1998. Informational complexity criteria for regression models. Computational Statistics & Data Analysis 28 (1), 51–76.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27, software available at `http://www.csie.ntu.edu.tw/ cjlin/libsvm`.

Christian, J., Gehl, A., Boudreaux, P., New, J., Dockery, R., 2010. Tennessee Valley Authoritys Campbell Creek Energy Efficient Homes Project: 2010 First Year Performance Report July 1, 2009August 31, 2010, 1–126.
URL `http://info.ornl.gov/sites/publications/files/Pub26374.pdf`

Crawley, D., Hand, J., Kummert, M., Griffith, B., Apr. 2008. Contrasting the capabilities of building energy performance simulation programs. Building and Environment 43 (4), 661–673.
URL `http://dx.doi.org/10.1016/j.buildenv.2006.10.027`

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39 (1), 1–38.

Dodier, R., Henze, G., 2004. Statistical analysis of neural networks as applied to building energy prediction. Journal of solar energy engineering 126, 592.

Dong, B., Cao, C., Lee, S., 2005. Applying support vector machines to predict building energy consumption in tropical region. Energy and Buildings 37 (5), 545–553.

Feuston, B., Thurtell, J., 1994. Generalized nonlinear regression with ensemble of neural nets: the great energy predictor shootout. ASHRAE Transactions. (5-1080).

Gonzalez, P., Zamarreno, J., 2005. Prediction of hourly energy consumption in buildings based on a feedback artificial neural network. Energy and buildings 37 (6), 595–601.

Hendron, R., Engebrecht, C., (US), N. R. E. L., 2010. Building America House Simulation Protocols. National Renewable Energy Laboratory.

Hoegaerts, L., Suykens, J., Vandewalle, J., De Moor, B., 2004. A comparison of pruning algorithms for sparse least squares support vector machines. In: Neural Information Processing. Springer, pp. 1247–1253.

Ibargüengoytia, P. H., Sucar, L. E., Vadera, S., Nov. 2001. Real Time Intelligent Sensor Validation. IEEE Transactions on Power Systems 16 (4), 770–775.

Iijima, M., Takeuchi, R., Takagi, K., Matsumoto, T., 1994. Piecewise-linear regression on the ashrae time-series data. ASHRAE Transactions 100 (2), 1088–1095.

Jordan, M., Jacobs, R., 1992. Hierarchies of Adaptive Experts. Advances in Neural Information Processing Systems 4, 985–993.

Jordan, M., Jacobs, R., 1994. Hierarchical Mixtures of Experts and the EM Algorithm. Neural computation 6 (2), 181–214.

Karatasou, S., Santamouris, M., Geros, V., 2006. Modeling and predicting building's energy use with artificial neural networks: Methods and results. Energy and buildings 38 (8), 949–958.

Kissock, J., Haberl, J., Claridge, D., 2003. Inverse modeling toolkit (1050rp): Numerical algorithms. ASHRAE transactions 109 (Part 2), 425–434.

Kolter, J., Ferreira Jr, J., 2011. A large-scale study on predicting and contextualizing building energy usage. In: Twenty-Fifth AAAI Conference on Artificial Intelligence.

Kreider, J., Haberl, J., 1994. Predicting hourly building energy use: the great energy predictor shootout- overview and discussion of results. ASHRAE Transactions 100 (2), 1104–1118.

Li, K., Su, H., Chu, J., 2011. Forecasting building energy consumption using neural networks and hybrid neuro-fuzzy system: a comparative study. Energy and Buildings.

Lima, C., Coelho, A., Von Zuben, F., 2009. Pattern classification with mixtures of weighted least-squares support vector machine experts. Neural computing & applications 18 (7), 843–860.

MacKay, D., et al., 1994. Bayesian nonlinear modeling for the prediction competition. Ashrae Transactions 100 (2), 1053–1062.

Martin, D., Fowlkes, C., Malik, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (5), 530–549.

Russell, S., Norvig, P., 2010. Artificial Intelligence: A Modern Approach. Prentice Hall series in artificial intelligence. Prentice Hall.
URL http://books.google.com/books?id=8jZBksh-bUMC

Schaller, R., Jun. 1997. Moore's law: past, present and future. IEEE Spectrum 34 (6), 52–59.
URL http://dx.doi.org/10.1109/6.591665

Smola, A.J. and Schólkopf, B., 2004. A tutorial on support vector regression. Statistics and computing 14 (3), 199–222.

Suykens, J., Gestel, T. V., Brabanter, J. D., Moor, B. D., Vandewalle, J., 2002a. Least squares support vector machines. World Scientific Pub Co Inc.

Suykens, J., J., D. B., L., L., J., V., 2002b. Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing 48 (1-4), 85–105.

U.S. Department of Energy, 2010. Buildings Energy Data Book. D&R International, Ltd., available at `http://buildingsdatabook.eren.doe.gov/`.

Vapnik, V., 1999. An overview of statistical learning theory. Neural Networks, IEEE Transactions on 10 (5), 988–999.

Yang, J., Rivard, H., Zmeureanu, R., 2005. On-line building energy prediction using adaptive artificial neural networks. Energy and buildings 37 (12), 1250–1259.