Sensor-Based Energy Modeling (sBEM) D4: Final Report

Richard Edwards and Lynne E. Parker
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville TN, USA
(redwar15,parker)@eecs.utk.edu

Prepared for:
Joshua New
Building Envelopes Research Group
and Whole Building Community Integration Group
Oak Ridge National Laboratory, Oak Ridge TN, USA
newjr@ornl.gov

September 30, 2011

Executive Summary

This document presents our results for predicting energy consumption using up to 144 channels of sensor data collected from three fully automated houses located at the Campbell Creek Subdivision of Knoxville, TN in coordination with the Tennessee Valley Authority (TVA) and Oak Ridge National Laboratory (ORNL). Using this sensor information, we apply ten different machine learning methods to predict current and future hourly electrical energy consumption. The machine learning algorithms tested on these home are Linear Regression (REG); Feed Forward Neural Networks (FFNN); Support Vector Regression (SVR); Least Squares Support Vector Machines (LSSVM); Hierarchical Mixtures of Experts (HME) with REG, FFNN, LSSVM Experts; and Fuzzy C-Means (FCM) with REG, FFNN, LSSVM Experts. We studied each learner's predictive capabilities on all three homes and show that data-driven models may be a viable alternative to complex simulation systems, such as the current Department of Energy (DOE) simulation standard, Energy Plus (E+), which require experts to configure and calibrate simulations for each building. Our results shows that LSSVM is the best machine learning technique for predicting next-hour electrical consumption on this data set. To the best of our knowledge, these models establish the first hourly prediction results for residential buildings.

We tested the aforementioned machine learning techniques on the American Society of Heating Refrigerating and Air Conditioning Engineer's (ASHRAE) Great Energy Prediction Shootout I competition commercial buildings data set, as well as two additional machine learning methods – Relevance Vector Machines (RVM) and Hidden Markov Experts (HiddenME) with FFNN Experts. The results validate the techniques in two ways. First, the learners are consistent with the existing literature (FFNNs are best for predicting commercial building electrical consumption). Second, the learners are implemented correctly since results do not differ greatly from the ASHRAE results. The Coefficient of Variance (CV) range for the prediction of the next hour's electrical consumption in our initial residential results are 20% to 30%, while commercial prediction results range from 8% to 12%.

In the near term, few buildings will have 100+ channels of sensor data so it is necessary to identify a subset of the most predictive sensors. We employed two model selection techniques, Stepwise Selection (SS) and a Genetic Algorithm (GA) with an Information Complexity (IC) objective function, which allow us to combine predictive accuracy, model complexity, and robustness in the subset selection process. Tests quantitatively verified that setting missing sensor data to zero was more effective than removing sensors with missing data, and allowing learning on 3 hours of previous data was better than Markov Order 1 or 2. Testing on each individual house and across all houses shows that the GA consistently identifies the best sensor subsets, and that it is possible to obtain better performance by combining its best models into a single model through a weighted voting mechanism. We employed brute-force computation to identify the best sensor subset from all possible subsets with sizes one through four (combination 144 choose 4 sensor sets tested). Comparing these "Restricted Ground Truth" subsets against our techniques quantifies the trade-off between machine learning approximations and the exact solution for the best subset, which is computationally infeasible for all but the smallest set of sensors.

Combining these machine learning algorithms and subset selection techniques allows us to address a key criticism against data driven methods – that each system is designed specifically for one target area or data set and is not re-deployable. It is anticipated that determining the best sensors and fully automated data-driven energy modeling methods will become a much more cost-effective solution than the business-as-usual modeling methods as additional channels of data become available via smart meter deployment, wireless sensor cost drops, and the increasing proliferation of smart devices.

Contents

List of Tables

List of Figures

1 Introduction

Residential and commercial buildings constitute the largest sector of US primary energy consumption at 40% [36]. 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 take as input parameters such as weather data, building geometry, envelope composition with material properties (e.g., thermal conductivity, specific heat, etc.), 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 [8].

"Inverse" modeling, on the other hand, takes as input known energy use and potentially other variables (e.g., typical or actual outdoor temperature). The software then uses a statistical model to estimate portions of energy expended for different purposes (e.g., heating or cooling) as well as potentially any of the inputs traditionally used for "forward" modeling.

Sensor-based energy modeling can be viewed as a hybrid of the "forward" and "inverse" modeling approaches. In this data-driven approach, sensor readings are the input and codify the state of the 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.

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 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 [2] 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 according to the same transistor density doubling every 18 months as defined by Moore's Law [30]. 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. This article shows that this problem can be addressed through selection of an optimal subset of 140 sensors for predicting hourly energy consumption for 3 residential buildings, but extrapolation across building types is unproven and sensor counts/types would 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 [15] 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 [29], rarely is there a known algorithm that is best for solving

a given problem (e.g., predicting next hour energy usage).

There are numerous sensor-based works which focus on predicting current and future electrical consumption for commercial buildings [21, 19, 22]. In addition, these studies have established which 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 studies conducted with residential buildings model monthly electrical consumption [20], 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 establish which 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 ten different machine learning techniques, and determining which ones are best for predicting future *hourly* 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 discusses related work in the area of sensor-based machine learning applied to building energy modeling; Section 3 provides a technical overview of the different machine learning algorithms we explore within this work; Section 4 presents a detailed description of the residential data set, experimental design, and evaluation criteria; Section 5 presents results for predicting future residential electrical consumption, as well as results that validate the machine learning algorithms' correctness; Section 6 presents sensor selection results; Section 7 provides discussion about the results; and Section 8 presents our conclusions and future directions.

2 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 [21].

The overall winner, [24], 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 irrele-

vant 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.

Another winner used Piecewise Linear Regression [16]. 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. This method is described in Section 3.6 in greater detail.

[12] 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.

A more recent wbe prediction study with commercial buildings uses Support Vector Machines (SVM) to predict monthly consumption [11]. Support Vector Machines 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 [37]. 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 3.3).

[19] 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 [22], presents results for the Energy Predictor Shoot 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.

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, [20] 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 4.1. 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.

3 Approach

We have tested 10 different machine learning techniques on our residential data sets, and we have tested 12 learners 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); Relevance Vector Machines (RVM); HME with Linear Regression, FFNN, and LS-SVM Experts; Fuzzy C-Means with Linear Regression, FFNN, LS-SVM; and Hidden Markov Experts (HiddenME) with FFNN.

3.1 Linear Regression

Linear Regression is the simplest technique, and can and provide a baseline performance. 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.

3.2 Feed Forward Neural Network

As mentioned previously, previous works have shown that Feed Forward Neural Networks (FFNN) are very capable at predicting electrical consumption. These previous works 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 works. 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 other prior works have shown good performance using this function [10, 39, 13, 19].

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.

3.3 Support Vector Regression

Support Vector Regression (SVR) was designed and developed to minimize structural risk [32]. 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:

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

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 [32] 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 [6].

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 advantage and disadvantages [32], 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.

3.4 Least Squares Support Vector Machine

Least Squares Support Vector Machines (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:

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

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 [33]. 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 several works that address the sparsity issue through pruning or via weighting the examples [34].

3.5 Relevance Vector Machine

Relevance Vector Machine (RVM) is a Bayesian approach to SVM. The method tries to find the most general linear model with the following form:

$$f(x) = \sum_{i=1}^{N} w_i K(x_i, x) + w_o$$

This is the same functional model used by SVM. However, the learning process and overall objective function used to learn the model is completely different. This method assumes a prior distribution over the weights, and uses this distribution for regularization and Auto Relevance Detection (ARD) [35]. The ARD method is the same method used by the overall winner for the Great Energy Prediction Shootout, which we discussed previously in Section 2.

The key advantage to this Bayesian approach is that the user specified hyper-parameters found in the SVM formulation are removed. However, the method still requires one to select the correct kernel function, K, and the appropriate parameter settings for that kernel

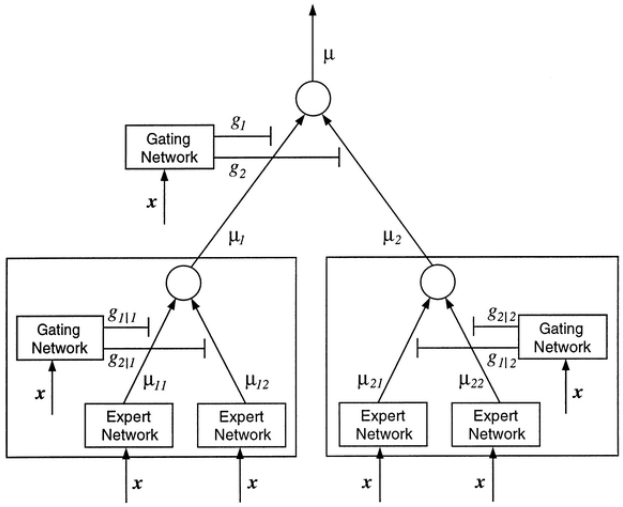


Figure 1: An example Hierarchical Mixture of Experts model with depth 2 and branching factor 2. This figure is provided by [18].

function. In addition, the RVM method learns the model by using EM. This methods EM learning process is well documented in [35]. However, EM is only guaranteed to converge to a locally optimal solution, while the SVM formulations guarantees a globally optimal solution given the hyper-parameters.

In this work, we only apply RVM to the Great Energy Prediction Shootout data set, because we were unable to find the appropriate kernel parameters that would allow the learner to converge on the residential data set. We are actively exploring methods for automatically determining the appropriate kernel function and kernel parameters; however, this problem is currently and open research topic.

3.6 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:

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

where g_i represents the top level gating network's output, $g_{j|i}$ represents the outputs from the lower level gating networks, and $F_{ji}(x)$ 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 works, the only supported expert learner was Neural Networks [17]; however, a later extension on the work introduced support for Linear Regression Experts [18]. While these works 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 works for regression problems is that each Expert's error rate follows a Gaussian distribution.

The original works 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:

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

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 were known, then maximizing the HME's likelihood function is split into two separate problems [18]. 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) [9] algorithms. The first algorithm is an exact EM algorithm, and the second algorithm approximates the first algorithm.

The EM-based learning algorithms relax the original Neural Network Expert restriction, because EM splits the learning process into two parts: Expectation and Maximization. The Expectation piece approximates $P(Z|X)$ for all Gating Networks, and the Maximization part approximates all parameters for the Experts. The Maximization process is presented as a weighted regression problem in both 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 [34] to integrate LS-SVM Experts in the HME framework. The robust LS-SVM algorithm estimates a weight for each training example and solves a weighted cost problem under the traditional LS-SVM framework. The training examples' weights are estimated based on traditional robust regression methods. However, we can substitute the weights generated by the EM algorithm for the robust LS-SVM weights, and solve the same weighted cost problem. This allows us to explore the standard FFNN and Linear Regression Experts, as well as LS-SVM Experts. In addition, HME with LS-SVM is a more general implementation of the Mixture of Experts with LS-SVM presented in [23]. That work integrated LS-SVM experts using a single Gating network, while we are able to support a hierarchy of mixtures.

3.7 Fuzzy C-Means with Local Models

An alternative approach to HME is to separate the learning process into two steps. The first step is an unsupervised learning phase which 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:

$$\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\|x_i - c_j\|^2$$

where u_{ij} represents the probability that x_i is a member of cluster c_j, and m is a user-defined parameter which controls how much an observation can belong to multiple clusters. The

criterion function is minimized through an iterative process using the following equations:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

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

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.

This work explores using the same previously mention experts, Linear Regression; FFNN; LS-SVM, for this two-step approach. In addition, the likelihood function requirement for these Experts are 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 $P(Y|X, \theta)$.

3.8 Hidden Markov Experts

Hidden Markov Experts (HiddenME) is very similar to HME with a single gating function and our FCM with local models approach. The key difference is that this method approximates $P(Z|X)$ by maximizing $P(Y|X, \theta, S)$, where S represents a set of unobservable states. This means the method assumes that the Experts change according to a Hidden Markov Model [26]. That is to say, the Expert that is responsible for the next prediction is dependent upon the Expert selected for the current prediction.

Under the Hidden Markov Model approach $P(Z|X)$ is represented as $P(S_t|X, S_t)$. This means that the previously presented $P(Z|X)$ is now approximated as the current belief distribution across the hidden states. The learning process for this approach uses EM and is documented in [38]. In addition, this approach requires that the user specifies the number of hidden states, as well as the initial transition probabilities between these states. The initialization for these parameters have very large impact on the overall learned model. In this work, we use random initialization for the transition parameters, and explore different numbers of hidden states. Lastly, we have not applied this method to the residential data set, because it requires that all learning examples are presented in chronological order. This makes it difficult to compare against the other methods, because our experimental design randomize the training examples, and without the randomization we may not obtain the best possible learners. We will address this issue by training all learners on the 2010 residential dataset, and test on the data currently being collected from the same residential homes for 2011.

3.9 Temporal Dependencies

In the realm of function approximation, *temporal dependencies* means that the target response y_t is dependent on past observations, x_{t-1}, as well as current observations 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 x_t, x_{t-1}, 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 x_{t-1}, 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 x_{t-1}, order 2 models use observations x_{t-1} and x_{t-2}, and so forth.

In previous works, 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; however, these methods present considerable research challenges and are beyond the scope of this article.

3.10 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. In addition, 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 predictive 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.

We use a entirely different approach to Model Selection for addressing sensor selection. We use a Model Selection technique called the Wrapper Model Selection method [14]. This method performs model selection by attempting to reduce the number of external parameters used by the learning system. Wrapper techniques provide a method for searching through different parameter configurations, using the given learning system to judge the quality of each configuration.

This concept is directly applicable to the sensor selection problem, in which the different parameter configurations are sensor subsets the learner can use to predict future energy consumption. In other words, one can restrict which sensors the learning system uses in the learning process, in order to determine which sensor subset produces the most general model.

3.11 Stepwise Selection for Sensor Selection

As mentioned previously, Stepwise Selection is a greedy search algorithm that attempts to minimize bias by only including parameters that contribute statistically significant improvements in performance. This process is carried out iteratively using two passes across the parameter space, where the first pass is a parameter inclusion step and the second pass is a parameter elimination step. The parameter inclusion pass starts by initializing an initial parameter set m, which is generally empty, and iterates over the parameter space in a fixed linear order $x_1, x_2, ...x_n$. At each iteration i, the algorithm tests to see if the current model m is statistically worse than the new model m' that includes parameter x_i. Model m and model m' are compared using the F-Test to either accept or reject the null hypothesis that parameter x_i does not increase model m's performance. If the null hypothesis is rejected with error confidence ρ, then the parameter x_i is added to the current model m.

The parameter elimination pass starts with model m after completing the parameter inclusion step, and iterates over the parameter space in the same fixed linear order. However, at each iteration i, the algorithm tests to see if the current model m is statistically better than model m'' that does not include parameter x_i. Model m and model m'' are compared using the same F-Test procedure, but the null hypothesis is now reformulated as m'' having worse performance than m. If there is not sufficient evidence to reject the null hypothesis with error confidence ρ', then parameter x_i is removed from model m.

The inclusion and elimination steps can be repeated until it is either no longer possible to add or remove a parameter from the subset, or for a fixed number of iterations if convergence is not possible. In this work, the Stepwise Selection procedure is performed until convergence, with ρ set to five percent and ρ' set to ten percent.

3.12 Model Criteria

There are many different Model Criteria functions that combine a goodness-of-fit objective with a model complexity objective. While each Model Criteria measures model complexity differently, all the functions measure goodness-of-fit using the same criteria, $-2\log(L(\theta))$, where $L(\theta)$ is the maximum likelihood function using θ as the parameter set. Since the Wrapper methods in this report use a Linear Regression Model as the learning system, the general maximum likelihood function should be expressed as follows:

$$L(\theta) = L(Y|\beta, \Sigma) = \frac{1}{2\pi^{k/2}|\Sigma|^{k/2}} e^{-\frac{(Y-X\beta)^{\mathrm{T}}\Sigma^{-1}(Y-X\beta)}{2}}$$

where k is the dimensionality of the multivariate normal (i.e., the number of parameters used in the regression model) and β is a coefficient matrix used to map the input X to a multivariate response Y. However, since our response variable Y (energy consumption) is univariate, the maximum likelihood equation simplifies to the following:

$$L(y|\beta, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{k/2}} e^{-\frac{(y-X\beta)^{\mathrm{T}}(y-X\beta)}{2\sigma^2}}$$

Given that all Model Criteria in this report are applied to univariate Linear Regression Models, one can replace $L(\theta)$ with $L(y|\beta, \sigma^2)$ to frame all Model Criteria for measuring regression complexity.

The first Model Criteria function was defined by Akaike in 1973, called AIC (Akaike's Information Criterion) [1]. This definition proposed the evaluation of a model based on the previous likelihood function $L(\theta)$ and a penalty term that attempts to correct the model's bias, under the assumption that the model that best minimizes $\log(L(\theta))$ and minimizes model complexity is the best model. AIC's Criteria function is as follows:

$$AIC(\theta) = -2\log(L(\theta)) + 2k$$

where k is the number of free parameters that are estimated in the model. After the introduction of AIC, many other Model Criteria functions were introduced, such as Bayesian Information Criterion [31], Minimum Description Length [27], Consistent AIC [3], and many more. [28] has illustrated that BIC, MDL, CAIC, and many other Model Criteria functions are able to find the true model, if a true model exists, or some approximate parsimonious model, otherwise. However, these methods only compute model complexity in terms of the number of estimated parameters, rather than also including the effect of parameter interactions.

Given that these previous Model Criteria functions compute model complexity without considering parameter interactions, we decided to use the Information Complexity (ICOMP) [5] Criteria. To the best of our knowledge, ICOMP is the only Model Criteria function that measures parameter interaction without the risk of under-fitting the model like CAICF([3]). The ICOMP Criteria function is defined as follows:

$$ICOMP(IFIM) = -2\log(L(\theta)) + 2C(F^{-1}(\theta))$$

where IFIM stands for Inverse Fisher Information Matrix and C is a specified complexity function that maps $F^{-1}(\theta)$, the estimated Inverse Fisher Information Matrix, under the parameters θ, to a single complexity score. Note that lower values of the ICOMP function are preferred. There are many different variants of ICOMP, each with a different C complexity function and each with a different approximation for $\Sigma(\theta)$ [4]. In this report, we make use of $ICOMP(IFIM)_{Misspec}$, since it is scale invariant, considers skewness and kurtosis within the model, and helps protect against over-fitting when the $L(\theta)$ function is incorrectly specified [4]. $ICOMP(IFIM)_{Misspec}$ is defined as follows:

$$ICOMP(IFIM)_{Misspec} = -2\log(L(\theta)) + 2C_1(Cov(\theta)_{Misspec})$$

where $Cov(\theta)_{Misspec}$ and $C_1(\Sigma)$ are defined as:

$$Cov(\theta)_{Misspec} = F^{-1}RF^{-1}$$

$$C_1(\Sigma) = \frac{p}{2}\log(\frac{tr(\Sigma)}{p}) - \frac{1}{2}|\Sigma|$$

Additionally, [4] illustrates that when applying $ICOMP(IFIM)_{Misspec}$ to regression models, F^{-1} and R are defined as:

$$F^{-1} = \begin{bmatrix} \sigma^2(X^{\mathrm{T}}X)^{-1} & 0 \\ 0 & \frac{2\sigma^4}{n} \end{bmatrix}$$

$$R = \begin{bmatrix} \frac{1}{\sigma^4}X^{\mathrm{T}}D^2X & X'1\frac{S_k}{2\sigma^3} \\ (X'1\frac{S_k}{2\sigma^3})' & \frac{(n-q)(K_t-1)}{4\sigma^4} \end{bmatrix}$$

where $D^2 = diag\{\varepsilon_1^2, ..., \varepsilon_n^2\}$ and ε_i^2 is the squared residual error for target y_i, X represents the input data to the regression model, S_k is skewness within the residual errors, and K_t is kurtosis.

3.13 Genetic Algorithm for Sensor Selection

A Genetic Algorithm solves a search problem by considering several candidate solutions in parallel and combining good solutions from the pool of candidate solutions to create new candidate solutions. The hope is that each time the algorithm creates new candidate solutions, they will be superior to the previous candidate solutions. This process is implemented through a set of fairly simplistic, but powerful, operations called selection, crossover, and mutation, which are performed on the current population, or candidate solution set, with respect to a user-defined fitness function that measures solution quality. A candidate solution for our Genetic Algorithm Wrapper for sensor selection is a binary string with a length equal to the number of sensors within the dataset; sensor x_i is included in the solution if the solution has a 1 at index i.

The selection operator determines which candidate solutions will enter the new population without modification and which solutions will be used for constructing new candidate solutions. This process can either uniformly select solutions from the population, select solutions according to a probability distribution derived from each solutions' quality, or select

according to a probability distribution defined over the current solution rankings. The latter option is used in this report.

The crossover operation uses the selection operator to pick two candidate solutions from the population and to probabilistically create either one or two candidate solutions. There are many different types of crossover operators; the method used in this report is called *scattered crossover*. This method selects two candidate solutions p_1 and p_2 from the population and generates a random binary string. The new candidate solution copies all elements from p_1 that correspond with a 1 in the binary string and all elements from p_2 that correspond with a 0 in the binary string.

Mutation uses the selection operation to pick a small percentage of the candidate solutions, roughly one or two percent, and then probabilistically determines if it should alter the selected candidate solutions. The alteration is based on a Bernoulli experiment performed on each binary bit of the selected candidate solutions. This means that with probability p, a single binary bit could change from 1 to 0 or vice versa. There is much controversy over whether or not mutation contributes to finding good candidate solutions, so p is generally set fairly small. In this particular application, p is set to 0.01.

A Genetic Algorithm combines these operators to optimize a fitness function, where the fitness function measures the quality for a candidate solution. In this particular application, we follow the work presented in [4], which suggests and illustrates using the previously mentioned $ICOMP(IFIM)$ measure as the fitness function, because of its previously stated beneficial properties.

3.14 Sensor Ranking

Since we are interested in finding which sensors are most useful for building a general energy prediction model, we can frame the problem as a model selection problem. However, each Wrapper method might produce a different best model answer when presented with different subsets of the original dataset. For example, if one uses 75% of a dataset for learning and the remaining 25% for testing purposes, the learning system can provide consistently different best models each time one resamples the data into learning and testing sets. This leaves two options — search for the best model among all possible best models or derive a method to combine the best models seen so far to construct a ranking for each selected sensor.

Option one is a viable option, because we are able to use $ICOMP(IFIM)$ to directly compare all seen best models, by selecting the model with the lowest $ICOMP(IFIM)$ score. However, there is an infinite number of best models, and it is not guaranteed that one will find the true best model. As will be seen in Section 6, the best model may not always have the smallest $ICOMP(IFIM)$ score, but rather the smallest variance. That is to say, the best sensor subset model will generally have a small variance over a wide range of different learning and testing sets.

Alternatively, one could devise a method for combining each model's best sensor subset through voting, since each model is a best model over some set of learning and testing configurations. In our opinion, the voting scheme for combining the best seen sensor models should be preferred for models with low $ICOMP(IFIM)$ scores, which have low variance

in addition to their low score. Therefore, our voting scheme is defined as follows:

$$v = \frac{ICOMP(IFIM)_{max} - ICOMP(IFIM)_m}{\sigma_m^2}$$

where v is model m's voting power, $ICOMP(IFIM)_{max}$ is the score for the worst sensor subset in the collection of seen models, and σ_m^2 is model m's $ICOMP(IFIM)$ variance. We then allow each model to cast a positive vote v for each sensor present in the model and a negative vote $-v$ for each sensor not present in the model. If we sum the votes for each sensor, we are able to assign a rank to each sensor based on the currently observed best model, by simply sorting all sensor final scores in descending order.

4 Methods

4.1 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 3 different homes located in west Knox County, Tennessee. In addition, 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 [7]. The first house in this study, called House 1, is a standard two-story residential home. However, the second, called House 2, and third house, called House 3, 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 two sets of photovoltaics – one set is for generating electricity, and a second is for heating water in a solar-powered water heater.

The key characteristic about this dataset is that each house has approximately 140 different sensors which collect data every 15 minutes, and that each house is 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. The simulated occupancy provides stable behavioral patterns across all three homes, making device usage within the dataset consistent across test environments. In addition, the homes are automated based on a DOE study of the typical energy usage patterns of American households. This means the data set is free from behavioral factors, making it easier to compare results for different houses. In addition, this data set provides a vast quantity of inputs that far surpasses the amount of information used in the commercial and residential building studies.

4.2 Prediction 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 hyper-parameters.

The different model configurations include testing: Markov order 1 through 3, different numbers of hidden neurons, different numbers of clusters, and different complete tree structures. For all HME models, we tested complete trees with depths 1 through 3 and branching factors 2 through 4. Every model that incorporates a FFNN was tested with 10 to 15 hidden neurons. Lastly, the Fuzzy Cluster approach was tested with 2 to 8 clusters. 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 [19, 22]. 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 previous known electrical consumption values.

4.3 Sensor Selection Experimental Design

Our primary interest is determining the most predictive sensor subset, rather than which Wrapper method is fundamentally better at selecting best models. However, given that each previously presented Wrapper method can produce completely different subsets of sensors, we must ultimately compare the two methods against each other when selecting the best model. We generate 200 unique models per Wrapper method for each house and across all houses, where 100 unique models are created from data with missing values set to zero, and the other 100 unique models are generated from data that had variables with missing values removed completely. We select four best models from each wrapper method by selecting two models from each respective 100 unique models. These two models are selected based on two different criteria – the first model selected has the lowest mean $ICOMP$ score, and the second selected model has the $ICOMP$ score with the least variance. We compare

the four best Genetic Algorithm sensor subsets against the four best Stepwise Selection sensor subsets, by testing each subset on 20 unique learning and test configurations sampled from each respective dataset. The learning and test configurations were generated using the Campbell Creek 2010 dataset, recorded from January 1st, 2010 until December 31st, 2010. 75% of the dataset is sampled without replacement for training and the remaining 25% is used as the testing set. We compare the eight selected best models found for each Campbell Creek house or across all houses — meaning that all the individual house datasets are combined into one dataset — and select the best performing model for each dataset.

Additionally, we apply our previously mentioned Sensor Ranking scheme to each set of 100 unique models, generated from each Wrapper method's 200 unique models, and compare the Sensor Rankings. We test to see which ranking is most general by selecting a fixed number of top sensors from each set and compare them using the same testing methodology for comparing the best selected models.

4.4 Performance Metrics

The primary measure for selecting the winners in the ASHRAE competition was the Coefficient of Variance (CV) measure [21], which determines how much the overall prediction error varies with respect to the target's mean. In other words, 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) [19, 13]. 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.

	S1			S2		
	CV(%)	MBE(%)	MAPE(%)	CV(%)	MBE(%)	MAPE(%)
Regression	14.12±0.00	7.69±0.00	13.41±0.00	4.07±0.00	1.01±0.00	2.86±0.00
FFNN	**11.29±0.00**	**8.32±0.00**	**9.14±0.00**	2.93±0.00	0.64±0.00	1.77±0.00
SVR	11.93±0.00	8.95±0.00	9.63±0.00	3.97±0.00	1.41±0.00	2.31±0.00
LSSVM	13.70±0.00	10.32±0.00	11.21±0.00	6.35±0.00	1.53±0.00	4.50±0.00
RVM	12.63±0.00	10.03±0.00	10.53±0.00	2.95±0.00	0.69±0.00	1.80±0.00
HME-REG	14.11±0.00	7.66±0.00	13.40±0.00	4.05±0.00	0.99±0.00	2.85±0.00
HME-LSSVM	13.61±0.00	10.21±0.00	11.13±0.00	6.33±0.00	1.61±0.00	4.51±0.00
HME-FFNN	11.49±0.00	2.91±0.00	9.73±0.00	**2.75±0.00**	**0.52±0.00**	**1.60±0.00**
FCM-REG	11.84±0.00	7.87±0.00	10.44±0.00	3.50±0.00	1.01±0.00	2.23±0.00
FCM-FFNN	11.51±0.00	8.71±0.00	9.45±0.00	**2.71±0.00**	**0.55±0.00**	**1.61±0.00**
FCM-LSSVM	13.47±0.00	10.36±0.00	11.04±0.00	6.59±0.00	1.55±0.00	4.78±0.00
HiddenME	**11.19±0.00**	**8.30±0.00**	**9.04±0.00**	**2.77±0.00**	**0.56±0.00**	**1.65±0.00**

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

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.

5 Prediction 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 section presents the best performing models from the ten 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.

5.1 Great Energy Prediction Shootout

For comparison purposes, we ran our 10 implemented machine learning techniques on the earlier Great Energy Prediction Shootout data set. In addition, we ran the previously mentioned RVM and HiddenME methods on this data set, as well. The 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 HiddenME and FFNN are the best predictors for electrical consumption. The difference between to the two is too small to conclude definitively which method is best. The FFNN finding is consistent with the existing literature [21]. However, all methods except Linear Regression, HME with Linear Regression, and all LSSVM based methods 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, FCM with FFNN, and HiddenME are better than the FFNN. However, the existing published results for the S2 inputs range from 2.44% to 3.65% [19, 22]. From these results, we can conclude that Neural Network type methods perform best on this data set. We can also conclude that the LSSVM

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
LSSVM	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-LSSVM	23.68±1.41	-0.03±0.99	18.69±0.85
HME-FFNN	22.77±1.56	0.15±0.98	17.74±0.65
FCM-REG	31.91±1.67	-0.09±0.91	29.74±0.86
FCM-FFNN	22.65±1.42	0.81±0.95	18.18±0.75
FCM-LSSVM	24.03±1.20	0.01±0.87	19.52±0.92

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
LSSVM	**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-LSSVM	20.23±0.85	0.07±0.56	16.40±0.80
HME-FFNN	20.15±1.65	0.46±0.93	17.07±1.19
FCM-REG	27.33±1.48	-0.14±0.72	25.62±0.80
FCM-FFNN	20.53±1.76	0.74±0.87	17.57±1.42
FCM-LSSVM	20.54±0.83	0.04±0.62	16.91±0.84

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
LSSVM	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-LSSVM	20.58±1.19	0.03±0.94	16.03±0.98
HME-FFNN	20.39±1.67	0.70±0.92	17.09±0.81
FCM-REG	26.33±1.72	-0.20±1.10	23.91±1.22
FCM-FFNN	21.03±1.29	0.47±1.49	18.27±1.06
FCM-LSSVM	20.50±1.47	0.07±0.69	16.11±1.15

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

based methods are the worst advanced technique, with Linear Regression and HME with Linear Regression being only slightly better.

5.2 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 LSSVM based methods, 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 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 the LSSVM based methods have the best MAPE measure and are statistically different from HME with FFNNs and FCM with FFNNs. Therefore, LSSVM is the best model for predicting next

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
LSSVM	**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-LSSVM	27.98±1.39	0.01±0.99	20.84±2.89
HME-FFNN	29.30±1.28	0.09±1.01	22.71±2.92
FCM-REG	35.20±0.87	0.05±1.99	29.77±2.41
FCM-FFNN	**28.14±1.21**	**0.40±0.97**	**21.96±2.74**
FCM-LSSVM	28.05±1.17	-0.03±1.00	21.01±2.33

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
LSSVM	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-LSSVM	27.63±1.28	0.10±0.89	20.41±3.42
HME-FFNN	28.17±2.04	0.26±0.58	22.43±2.44
FCM-REG	33.49±1.52	0.01±1.59	27.46±2.77
FCM-FFNN	28.34±1.67	-0.20±1.27	22.30±3.28
FCM-LSSVM	27.19±1.90	0.16±1.14	20.17±2.34

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
LSSVM	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-LSSVM	27.19±1.87	0.37±0.84	20.67±2.30
HME-FFNN	29.64±2.21	-0.12±1.64	24.81±0.38
FCM-REG	32.70±1.66	-0.00±2.02	27.12±2.91
FCM-FFNN	28.94±1.46	0.45±1.27	22.76±2.03
FCM-LSSVM	27.24±1.93	-0.01±1.76	19.70±2.53

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

hour energy consumption for House 1, because HME with LSSVM and FCM with LSSVM are not statistically different from a stand alone LSSVM.

5.3 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 the Order 1 LSSVM and Order 1 FCM with FFNNs. The HME with LSSVM and FCM with LSSVM are not selected, because their performance is not statistically different from from a stand alone LSSVM. In addition, 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 increase, 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.

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
LSSVM	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-LSSVM	31.85±1.83	0.14±1.12	23.03±2.48
HME-FFNN	32.98±1.28	-0.12±0.84	23.99±1.63
FCM-REG	39.69±3.11	0.12±1.30	31.58±1.88
FCM-FFNN	33.03±1.67	0.93±1.52	25.28±2.14
FCM-LSSVM	31.75±2.01	-0.12±1.09	22.76±1.29

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
LSSVM	**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-LSSVM	30.61±2.23	-0.25±1.74	21.22±1.34
HME-FFNN	32.99±2.17	1.07±1.17	24.76±1.94
FCM-REG	38.74±2.67	0.08±1.90	30.56±1.76
FCM-FFNN	32.92±2.49	0.76±2.03	24.20±2.06
FCM-LSSVM	30.48±2.39	-0.04±0.99	21.24±1.36

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
LSSVM	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-LSSVM	30.97±1.37	-0.21±0.97	21.37±1.61
HME-FFNN	33.34±1.83	1.09±1.24	25.15±2.13
FCM-REG	37.66±1.88	0.04±1.06	29.82±1.67
FCM-FFNN	33.66±2.09	1.17±1.30	25.51±1.72
FCM-LSSVM	30.57±2.55	-0.19±1.02	21.22±1.58

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

5.4 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 LSSVM, HME-LSSVM, and FCM-LSSVM. The Order 3 SVR shows improvement in the CV measure, while the Order 2 LSSVM based methods 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 LSSVM's, CV values. In addition, order 1 LSSVM' CV value is not statistically different from all HME with FFNN models and FCM with FFNN models, but the CV values for Order 2 and 3 are statistically better. Therefore, order 2 LSSVM and order 3 SVR are the best models based on the CV measure. The order 3 LSSVM model is excluded because it is not statistically different from the simpler order 2 model. Additionally, Order 2 HME with LSSVM and FCM with LSSVM are excluded, because they are not statistically different from the standalone order 2 LSSVM 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 LSSVM.

5.5 Prediction 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 [22].

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 LSSVM based methods work well on all three houses. However, LSSVM is statistically the best technique at predicting future residential electrical consumption over the next hour.

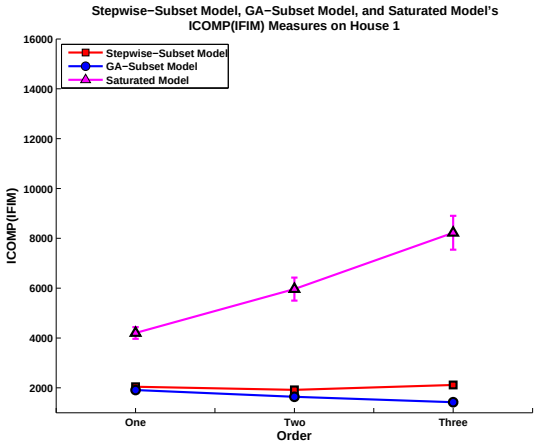
6 Sensor Selection Result

We have organized our sensor selection results according to the following order: Campbell Creek House 1, Campbell Creek House 2, Campbell Creek House 3, Across All Houses, Variable Ranking results, and comparisons against Ground Truth. The individual house sections and the Across All Houses section contain results generated from the selected eight best models. The Variable Ranking section contains results from applying our sensor ranking method mentioned in Section 3.14. The Ground Truth Comparison section presents the results from comparing the best sensors subsets with sizes one through four against the best Markov Order 1 models and the best top 10 sensor sets selected using our ranking method.

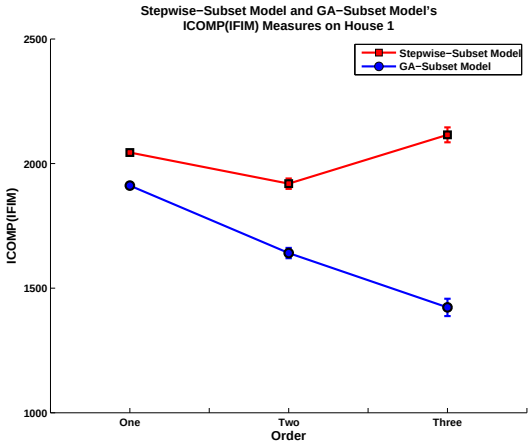
6.1 Campbell Creek House 1

Figure 2 illustrates the experimental results of comparing the Genetic Algorithm and Stepwise Selection Wrappers based on lowest $ICOMP(IFIM)$ variance, for Campbell Creek House 1. In addition, variables that have missing values were dropped, leaving each method with 87 candidate sensors. Under this particular best model selection, Figure 2 shows that the Genetic Algorithm Wrapper finds a more general subset of sensors for Markov Orders 1, 2, and 3. Interestingly, the model selected by the Genetic Algorithm uses more parameters than the model selected with Stepwise Selection for all Markov Orders. The Genetic Algorithm subset uses 57 sensors for Markov Order 1, 69 sensors for Markov Order 2, and 80 sensors for Markov Order 3, while the Stepwise Selection model uses 48 sensors, 58 sensors, and 69 sensors, respectively. This means that the Genetic Algorithm finds sensors it can incorporate without increasing the model complexity, while still producing a slightly better goodness-of-fit as the Stepwise Selection Wrapper (Figure 2(e)).

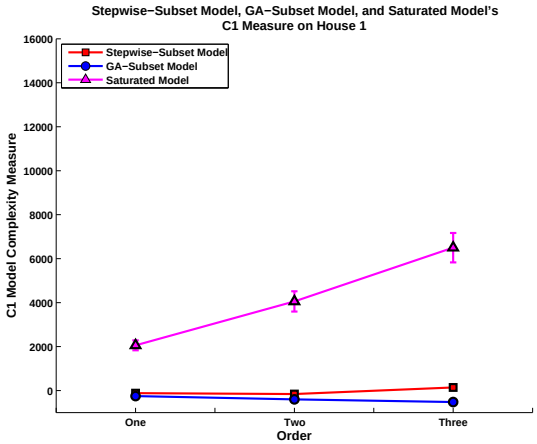
If we change the best model selection policy for the Campbell Creek House 1 dataset with the same 87 candidate sensors to selecting the model with the lowest mean $ICOMP(IFIM)$,



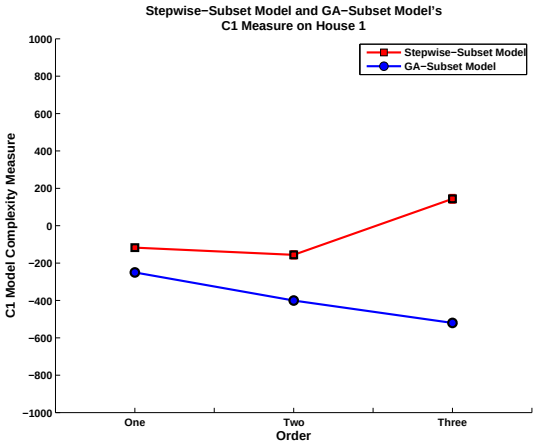
(a) Saturated Model's ICOMP



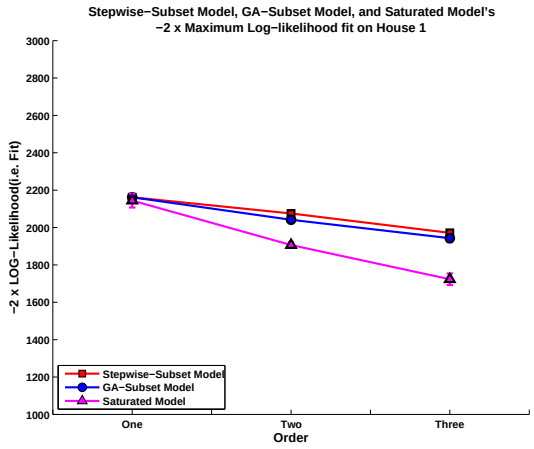
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity



(d) GA and Stepwise Model Complexity



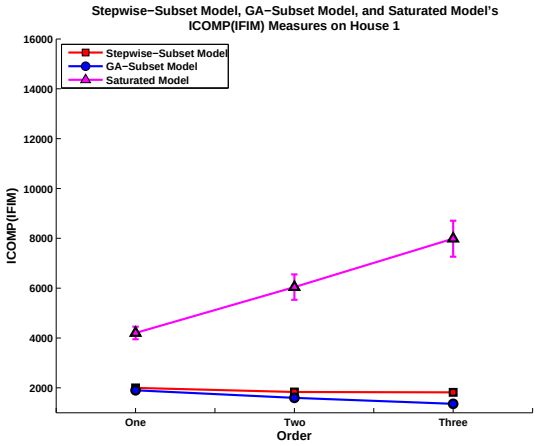
(e) Goodness-of-Fit

Figure 2: These graphs illustrate the experimental results from applying the models with the lowest $ICOMP(IFIM)$ variances on Campbell Creek House 1. Variables with missing data were removed from the dataset for these results.

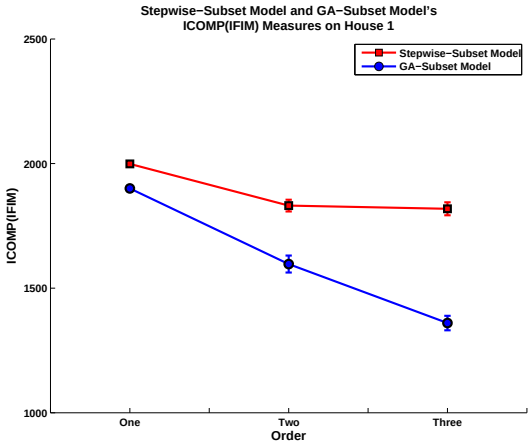
then the Genetic Algorithm method shows slight improvement in overall $ICOMP(IFIM)$ criteria, and the Stepwise Selection method's overall $ICOMP(IFIM)$ improves greatly for Orders 2 and 3. However, the goodness-of-fit (Figure 3(e)) for Genetic Algorithm methods show improvements over the best variance model (Figure 2(e)), while the Stepwise Selection Method shows degradation in performance. It may not be statistically different with a 95% confidence, but the Genetic Algorithm method appears to fit the data better in higher orders. While the overall $ICOMP(IFIM)$ criteria mostly improves, note a slight increase in the overall error range, meaning that these models are possibly more variable than best variance models. This means that when selecting the appropriate sensor subset one needs to consider the possible variance in performance in addition to overall performance. The Stepwise Selection method increases the number of sensors it selects for Markov Orders 1 and 2 in this set of experiments, using 50 sensors, 62 sensors, and 68 sensors. Additionally, the Genetic Algorithm method increases the number of sensors included in Markov Order 1 and 2. It uses 58 sensors, 73 sensors, and 78 sensors for these results.

Using the data from the same house, except that missing values are now set to zero and the number of candidate sensors is now 95, Figure 4 compares results for the Genetic Algorithm and Stepwise Selection methods based on the model with the lowest $ICOMP(IFIM)$ variance. Under these new conditions, the Genetic Algorithm's $ICOMP(IFIM)$ values are significantly worse than the two models selected when dropping variables with missing values (Figure 2(b) and Figure 3(b)). Similarly, the Stepwise Selection method's $ICOMP(IFIM)$ values are significantly worse for Markov Orders 1 and 2, but its $ICOMP(IFIM)$ value for Markov Order 3 is substantially better. It is not quite clear why this Stepwise Selection model performs better than the previous Stepwise Selection models for only Order 3. It could be because the overall fit is better when compared to the previous Stepwise Selection models, and the complexity is higher for order 1 and 2 causing the model to incur an additional penalty for the improved fit. The Genetic Algorithm's fit is significantly worse than the previous lowest $ICOMP(IFIM)$ mean value model (Figure 3(b)), yet there is only a slight degradation when compared to the previous lowest variance $ICOMP(IFIM)$ (Figure 2(b)). The Genetic Algorithm model used 57 sensors, 73 sensors, and 77 sensors and the Stepwise Selection model used 54 sensors, 66 sensors, and 73 sensors.

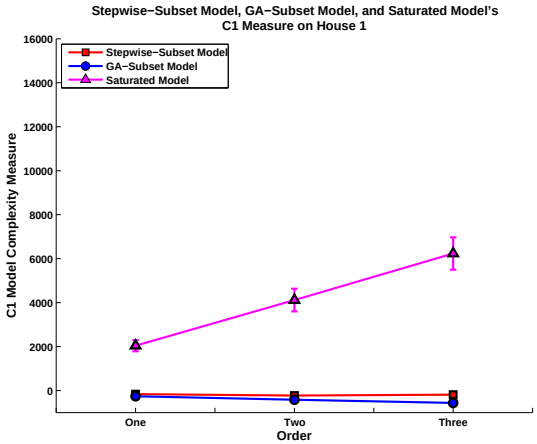
If we change the best model selection policy for the Campbell Creek House 1 dataset with the same 95 candidate sensors to selecting the model with the lowest mean $ICOMP(IFIM)$, then the Genetic Algorithm model's $ICOMP(IFIM)$ values (Figure 5(b)) are much closer to Genetic Algorithm results seen in Figures 2(b) and 3(b), while the Stepwise Selection model's $ICOMP(IFIM)$ values are the best results for this model selection on Campbell Creek House 1. The fit for the Genetic Algorithm model, Figure 5(e), is slightly better than the models seen in Figure 2(e) and Figure 4(e), but is worse than the Genetic Algorithm model in Figure 3(e). The Stepwise selection model's fit is mostly identical to the fit seen in Figure 2(e). This means the Stepwise Selection model is using sensors that were originally removed from the dataset, and these sensors provide improvement by reducing model complexity. This Stepwise Selection model increased the number of sensors used for Markov Order 1 and 2, compared to the lowest $ICOMP(IFIM)$ variance Stepwise Selection model. It uses 59



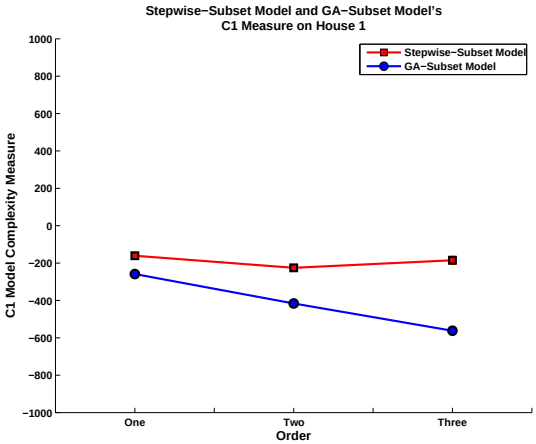
(a) Saturated Model's ICOMP



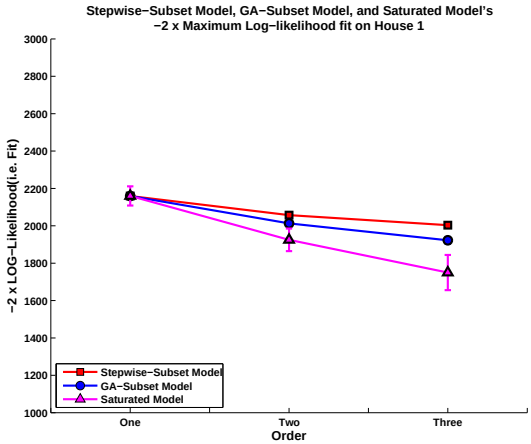
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

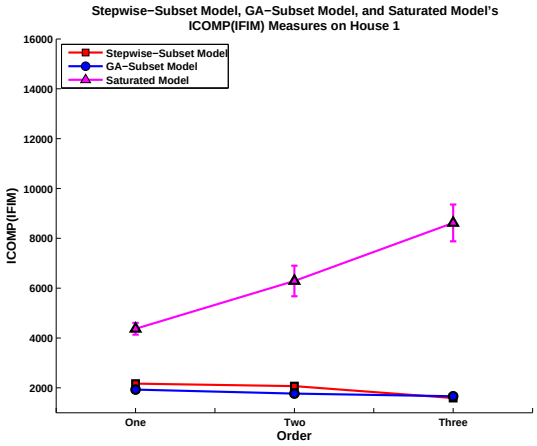


(d) GA and Stepwise Model Complexity

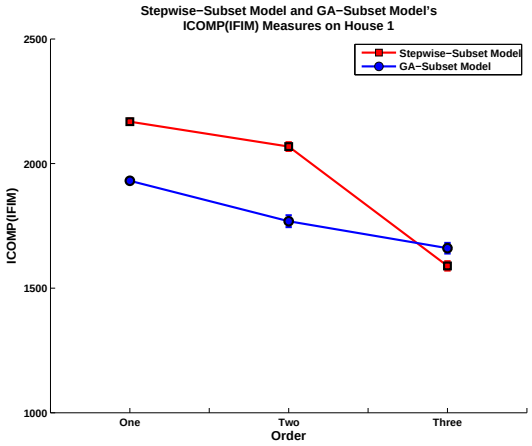


(e) Goodness-of-Fit

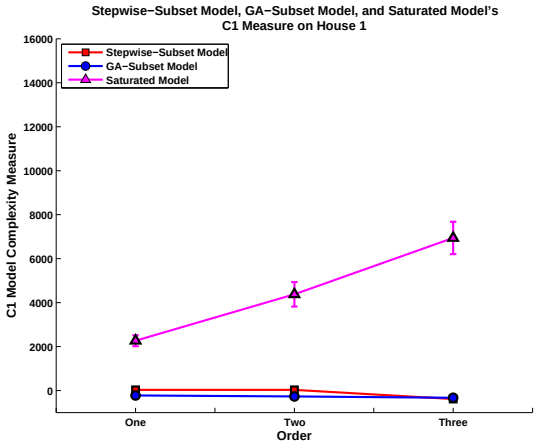
Figure 3: These graphs illustrate the experimental results from applying the models with the lowest mean $ICOMP(IFIM)$ on Campbell Creek House 1. Variables with missing data were removed from the dataset for these results.



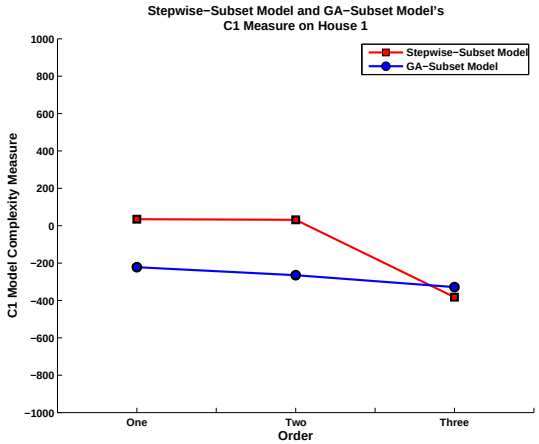
(a) Saturated Model's ICOMP



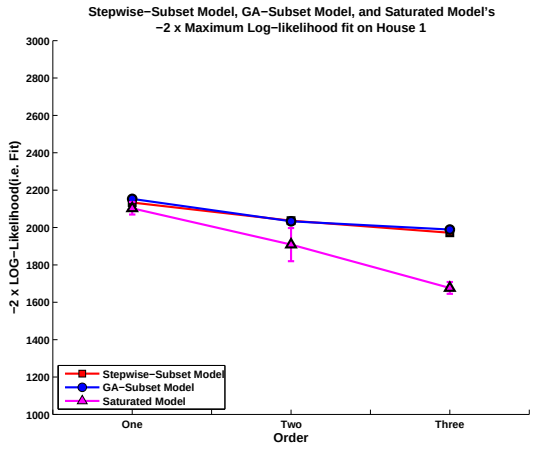
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity



(d) GA and Stepwise's Model Complexity



(e) Goodness-of-Fit

Figure 4: These graphs illustrate the experimental results from applying the models with the lowest $ICOMP(IFIM)$ variance on Campbell Creek House 1. All missing values in the data were set to zero for these results.

sensors, 71 sensors, and 73 sensors, while the Genetic Algorithm model uses 58 sensors, 72 sensors, and 89 sensors.

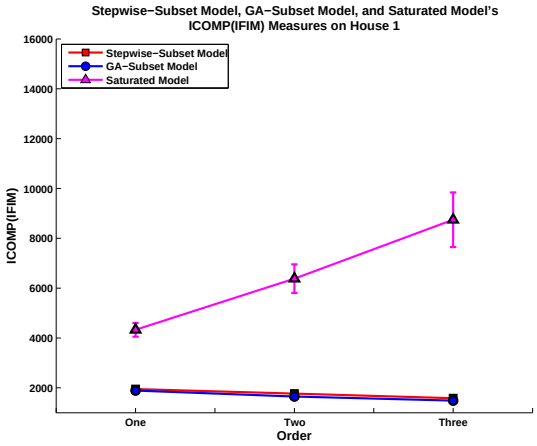
From Figures 2, 3, 4, and 5, it is clear that the best Genetic Algorithm Model for Campbell Creek House 1 is the model presented in Figure 3, and the best Stepwise Selection Model is presented in Figure 5. The dropped variables had a very large impact on the Stepwise Selection method, making it very difficult to find good models under the $ICOMP(IFIM)$ criteria. However, the Genetic Algorithm method in both cases was able to find better models than the Stepwise Selection method, but its best model was found when the variables with missing values were dropped. Ultimately, the Genetic Algorithm method is finding better models than Stepwise Selection on Campbell Creek House 1.

6.2 Campbell Creek House 2

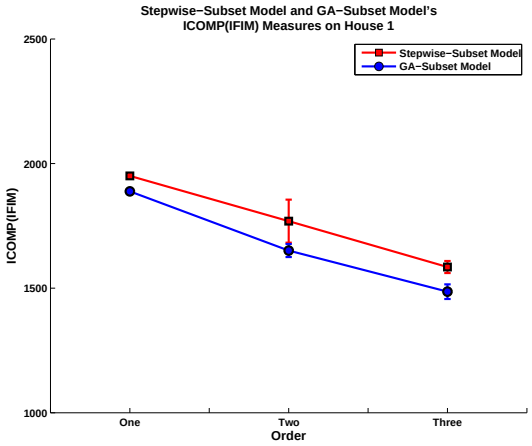
Figure 6 compares the results of the Genetic Algorithm and Stepwise Selection Wrappers when selecting the best model, based on lowest $ICOMP(IFIM)$ variance, for Campbell Creek House 2. In addition, variables that have missing values were dropped, leaving each method with 84 candidate sensors. Note that under the lowest variance model selection, the Genetic Algorithm finds a better model under the $ICOMP(IFIM)$ metric on this dataset, too. The Genetic Algorithm model uses considerably more sensors for all Markov Orders — 57 sensors, 67 sensors, and 73 sensors, while the Stepwise Selection model uses 46 sensors, 54 sensors, and 53 sensors. The differences in the numbers of sensors explains why the Genetic Algorithm model has a slightly better goodness-of-fit than the Stepwise Selection model (Figure 7(e)), because additional sensors included in the model can only increase goodness-of-fit; this is demonstrated with the fully saturated model (Figure 7(e)) where a fully saturated model is defined as one that uses all available sensors.

Changing the best model selection strategy to selecting the model with the lowest mean $ICOMP(IFIM)$ increases overall performance on the Campbell Creek House 2 dataset with 84 candidate sensors for the model generated using Stepwise Model Selection (Figure 7(b)). The Genetic Algorithm model presents very minor improvements for Markov Order 2 and 3. This stems from the Genetic Algorithm's goodness-of-fit (Figure 7(e)) and model complexity (Figure 7(d)) not significantly changing because the number of sensors included in the model remains roughly the same, as the best variance model. The Genetic Algorithm model, in Figure 7, uses 60 sensors, 69 sensors, and 73 sensors. Conversely, the Stepwise Selection model's goodness-of-fit appears to increase very slightly. The goodness-of-fit's means are shifted slightly lower than the original means (Figure 7(e) and Figure 6(e)). The increase stems from the Stepwise Selection method adding additional sensors to the model, using 51 sensors, 62 sensors, and 60 sensors.

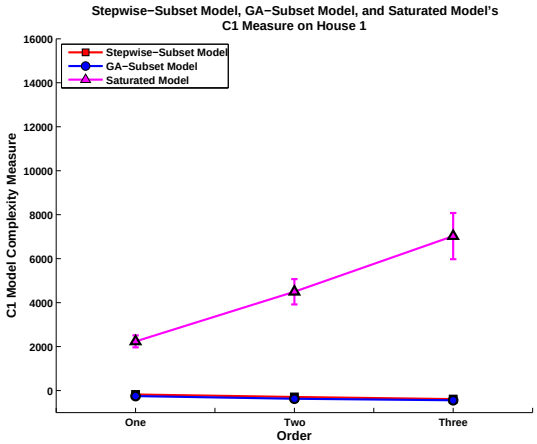
Using the data from the same house, except missing values are now set to zero and the number of candidate sensors is now 103, Figure 8 compares results for the Genetic Algorithm and Stepwise Selection methods based on the model with the lowest $ICOMP(IFIM)$ variance. The Genetic Algorithm's overall $ICOMP(IFIM)$ scores show decreases in performance, when compared to results shown in Figure 7(b) and Figure 6(b). The overall $ICOMP(IFIM)$ scores for Stepwise Selection are slightly better than the results seen in



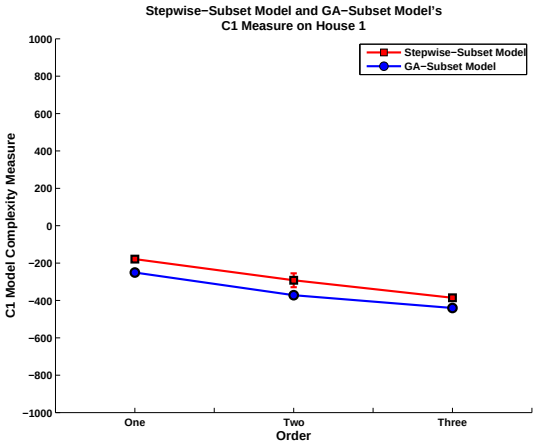
(a) Saturated Model's ICOMP



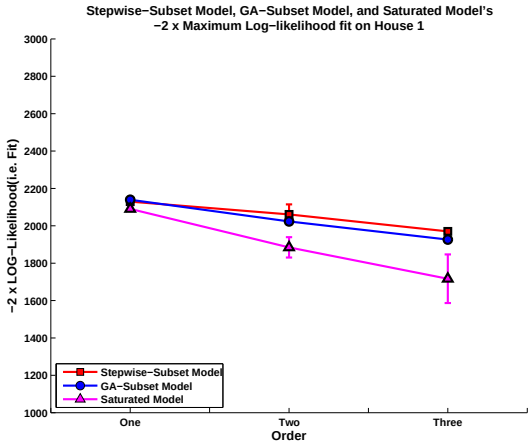
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

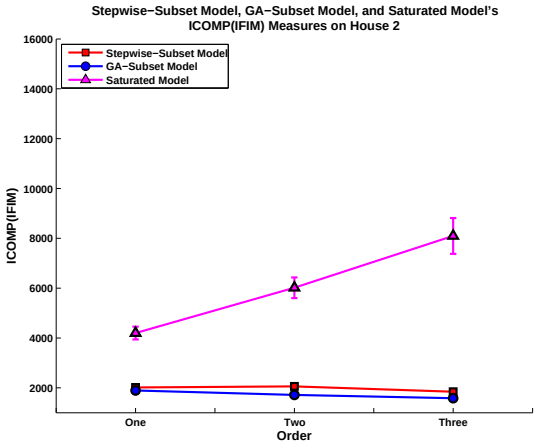


(d) GA and Stepwise's Model Complexity

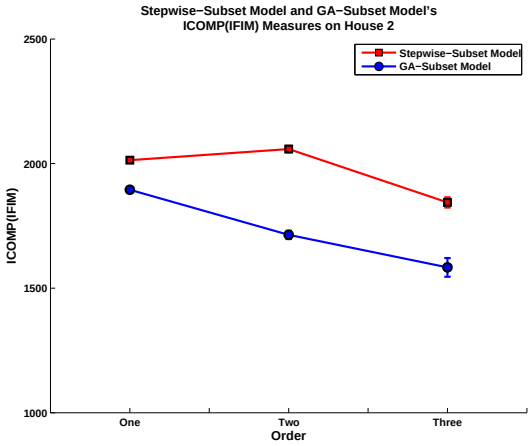


(e) Goodness-of-Fit

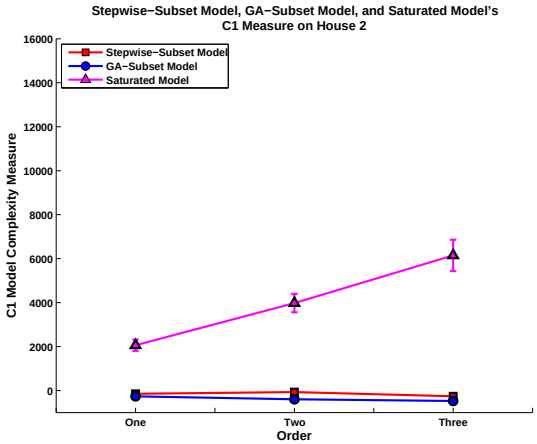
Figure 5: These graphs illustrate the experimental results from applying the models with the lowest mean $ICOMP(IFIM)$ on Campbell Creek House 1. All missing values in the data were set to zero for these results.



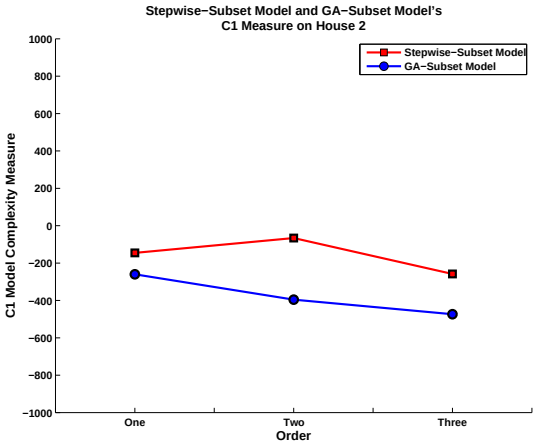
(a) Saturated Model's ICOMP



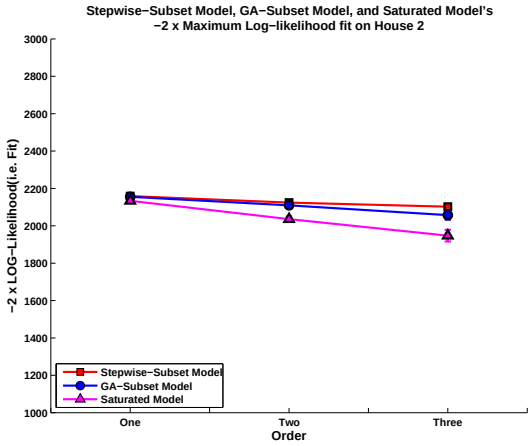
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

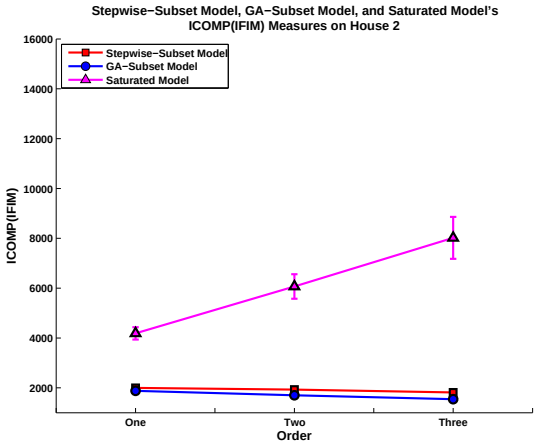


(d) GA and Stepwise's Model Complexity

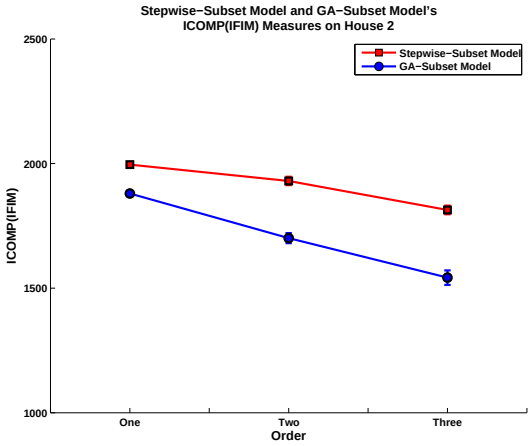


(e) Goodness-of-Fit

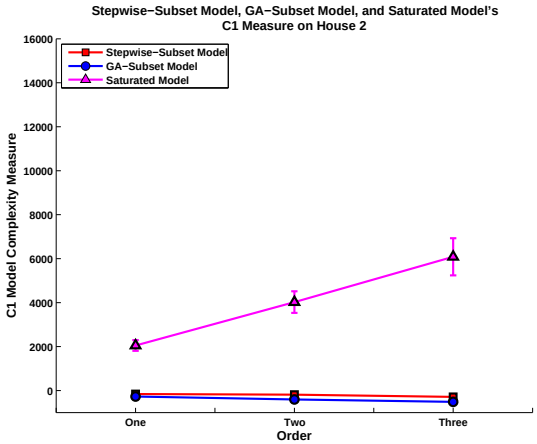
Figure 6: These graphs illustrate the experimental results from applying the models with the lowest $ICOMP(IFIM)$ variances on Campbell Creek House 2. Variables with missing data were removed from the dataset for these results.



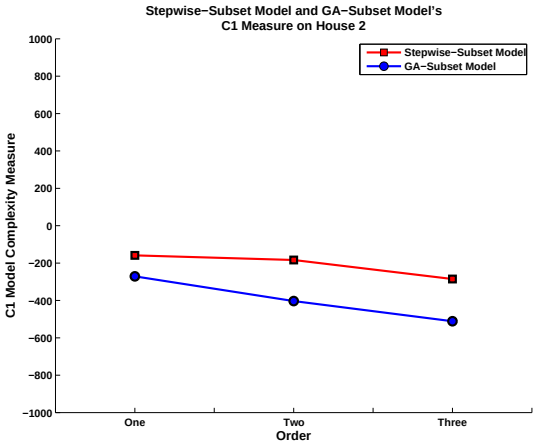
(a) Saturated Model's ICOMP



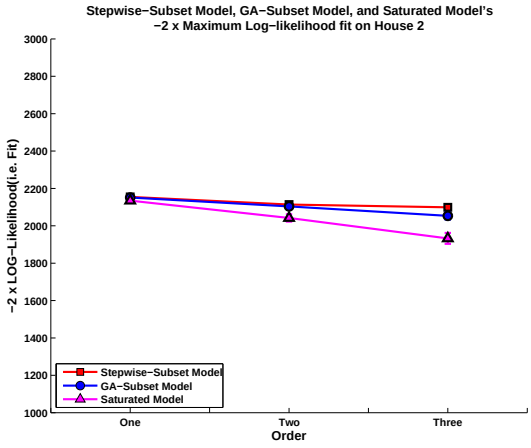
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity



(d) GA and Stepwise's Model Complexity



(e) Goodness-of-Fit

Figure 7: These graphs illustrate the experimental results from applying the models with the lowest mean $ICOMP(IFIM)$ on Campbell Creek House 2. Variables with missing data were removed from the dataset for these results.

Figure 6(b), but are considerably worse than the results shown in Figure 7(b). Additionally, the goodness of fit is slightly worse for both the Genetic Algorithm and Stepwise Selection compared to the previous models. The Genetic Algorithm uses 67 sensors for Markov Order 1, 85 sensors for Markov Order 2, and 91 sensors for Markov Order 3, while the Stepwise Selection method uses 63 sensors, 62 sensors, and 67 sensors.

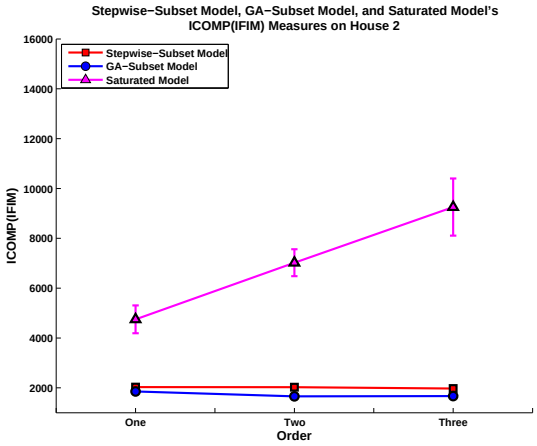
Changing the best model selection strategy to selecting the model with the lowest mean $ICOMP(IFIM)$ increases overall performance on the Campbell Creek House 2 dataset with 103 candidate sensors. The models generated using Stepwise Model Selection for Markov Orders 1 and 3 (Figure 9(b)) perform better than all other Stepwise Models on Campbell Creek House 2. However, its performance for Order 2 remains essentially the same as all other Stepwise Models. Additionally, the Genetic Algorithm method finds the best performing model in terms of $ICOMP(IFIM)$, compared to the other models presented in Figures 7(b), 6(b), and 8(b). The Genetic Algorithm method uses 69 sensors, 78 sensors, and 93 sensors. The best performing Stepwise Selection method uses 61 sensors, 62 sensors, and 68 sensors.

From Figures 6, 7, 8, and 9, it is clear that the best Genetic Algorithm Model for Campbell Creek House 2 is the model presented in Figure 9, and the best Stepwise Selection Model is presented in Figure 9. Similar to House 1, the dropped variables had a very large impact on the Stepwise Selection method, making it very difficult to find good models under the $ICOMP(IFIM)$ criteria. However, the Genetic Algorithm method in both cases was able to find better models than the Stepwise Selection method, but its best model was found when the variables with missing values were dropped. Ultimately, the Genetic Algorithm method is finding better models than Stepwise Selection on Campbell Creek House 2.

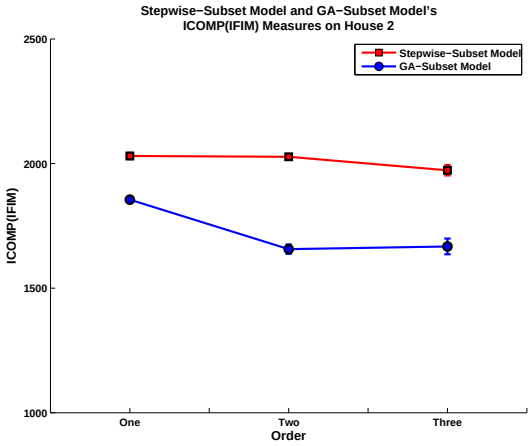
6.3 Campbell Creek House 3

Figure 10 compares the results of the Genetic Algorithm and Stepwise Selection Wrappers when selecting the best model based on lowest $ICOMP(IFIM)$ variance on Campbell Creek House 3. In addition, variables that have missing values were dropped, leaving each method with 77 candidate sensors. Recall that House 3 is the house for which a linear regression technique is not able to obtain a near-perfect mapping from x_t to y_t, while these mappings were successfully found for Houses 1 and 2. With this in mind, note that the $ICOMP(IFIM)$ scores are considerably higher (and thus worse) compared to the ones seen for Houses 1 and 2. Additionally, for all Markov Orders, the model selected with the Genetic Algorithm is better in terms of $ICOMP(IFIM)$, and model complexity (Figure 10(b) and Figure 10(d)), but the goodness-of-fit is essentially the same as the Stepwise Selection model (Figure 10(e)). The model selected by the Genetic Algorithm uses 41 sensors, 48 sensors, and 56 sensors, while the model selected by Stepwise Selection uses 49 sensors, 52 sensors, and 53 sensors.

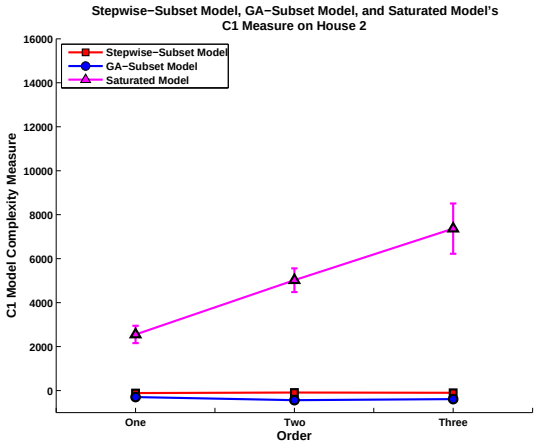
Changing the best model selection strategy to one of selecting the model with the lowest mean $ICOMP(IFIM)$ value on the House 3 dataset, with 77 candidate sensors, shows improvement for Markov Order 3 in term of $ICOMP(IFIM)$ values for both methods, but little to no increase for goodness-of-fit. Figures 10 and 11 strongly suggest that a different



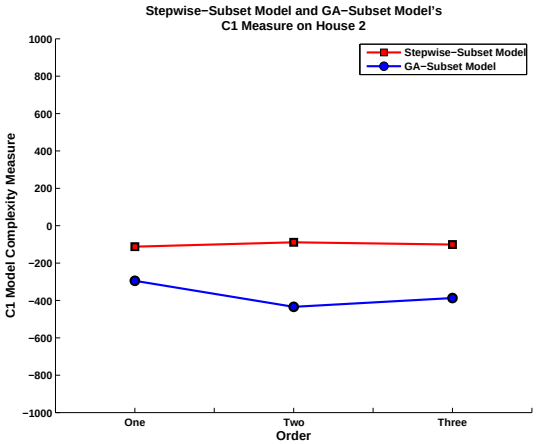
(a) Saturated Model's ICOMP



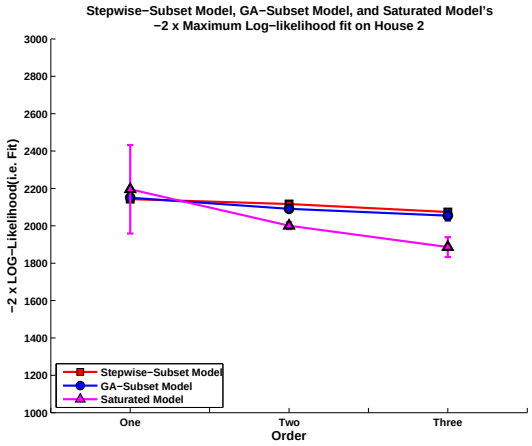
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

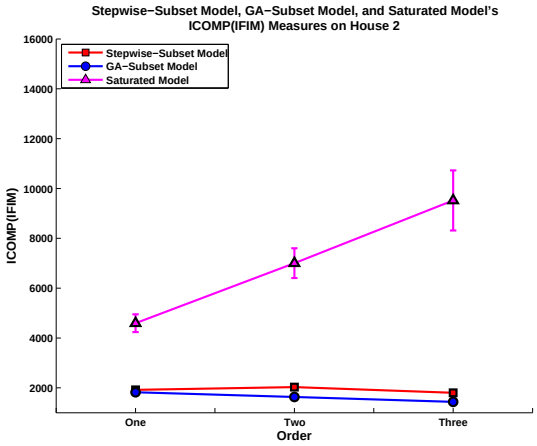


(d) GA and Stepwise Models' Complexity

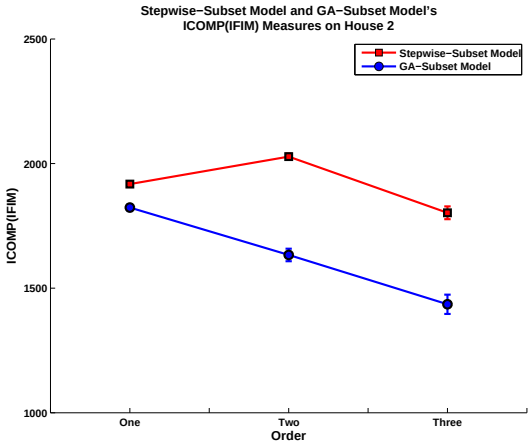


(e) Goodness-of-Fit

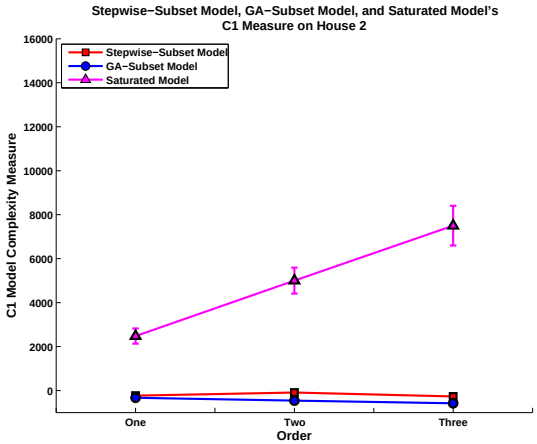
Figure 8: These graphs illustrate the experimental results from applying the models with the lowest $ICOMP(IFIM)$ variance on Campbell Creek House 2. All missing values in the data were set to zero for these results.



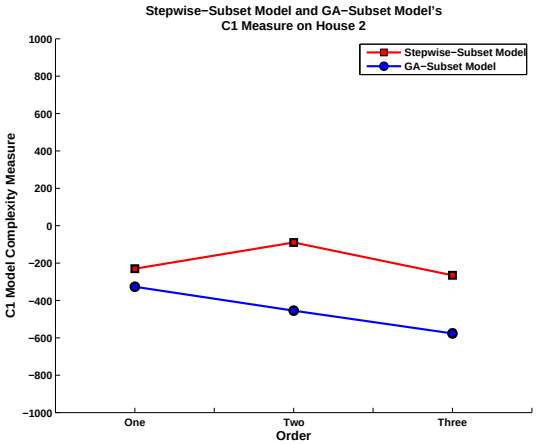
(a) Saturated Model's ICOMP



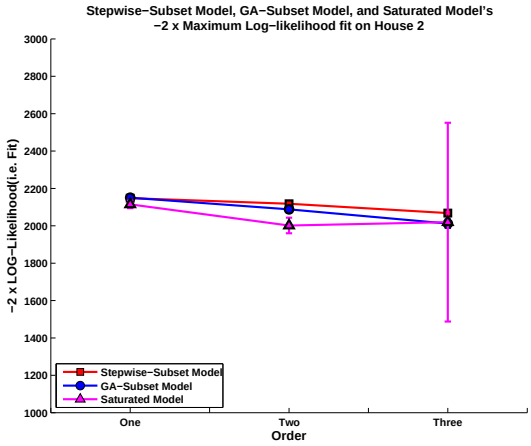
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

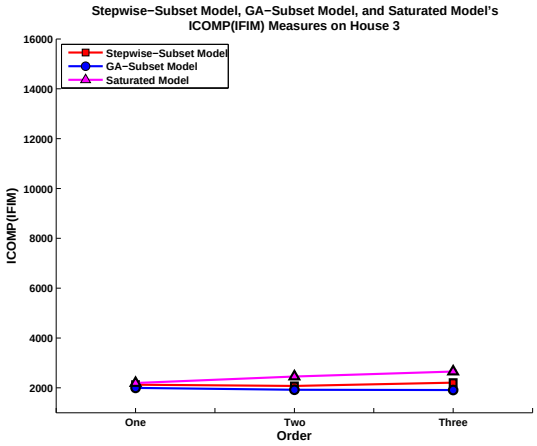


(d) GA and Stepwise Models' Complexity

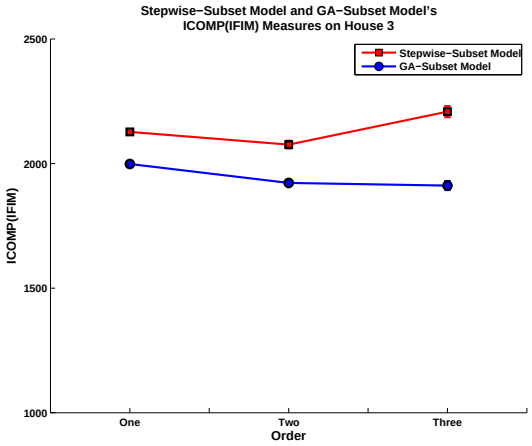


(e) Goodness-of-Fit

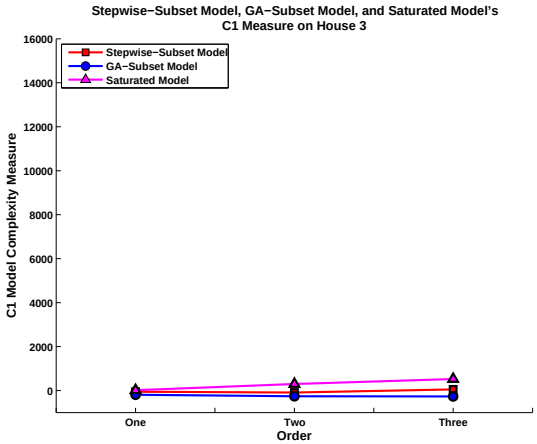
Figure 9: These graphs illustrate the experimental results from applying the models with the lowest mean $ICOMP(IFIM)$ on Campbell Creek House 2. All missing values in the data were set to zero for these results.



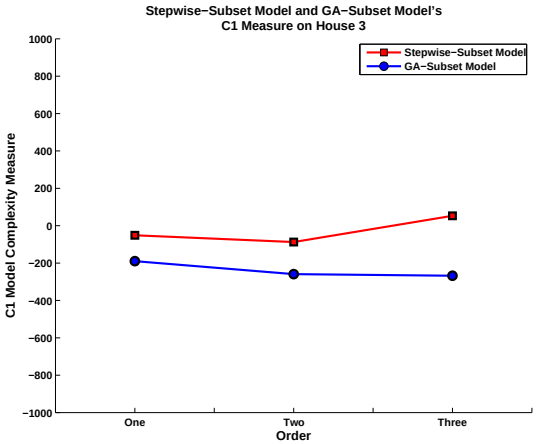
(a) Saturated Model's ICOMP



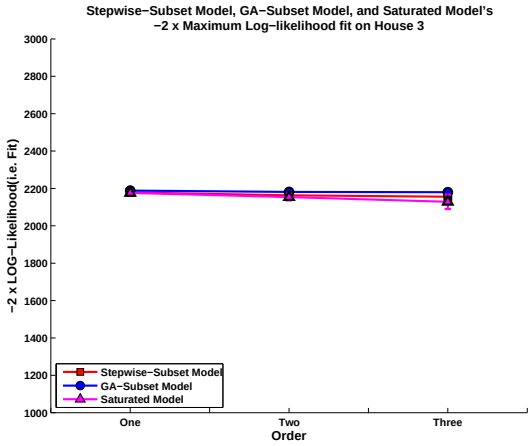
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity



(d) GA and Stepwise Models' Complexity



(e) Goodness-of-Fit

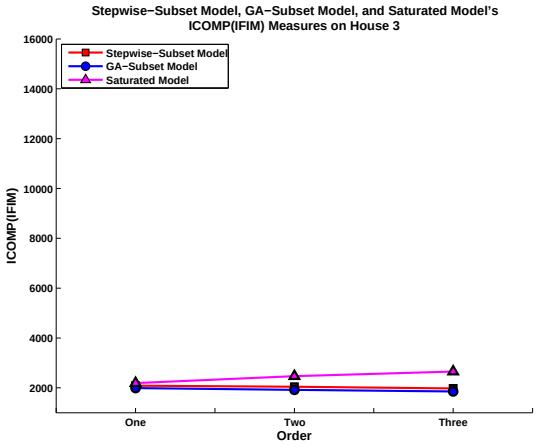
Figure 10: These graphs illustrate the experimental results from applying the models with the lowest $ICOMP(IFIM)$ variances on Campbell Creek House 3. Variables with missing data were removed from the dataset for these results.

approach is required for modeling House 3, because the overall model complexity for the fully saturated model is extremely low (Figure 10(c)) when compared to the overall model complexity for the fully saturated model on Houses 1 and 2 (Figure 2(c) and Figure 8(c)). This argues that there are complex non-linear relationships between House 3's sensor data and the actual energy consumption; we currently believe this difference stems from the fact that House 3 has the capability to produce a portion of its own electricity using solar panels. However, one can clearly see that the Stepwise Selection and Genetic Algorithm methods still minimize the selected model complexity, even though a Linear Regression Model may not be the most appropriate Learning method.

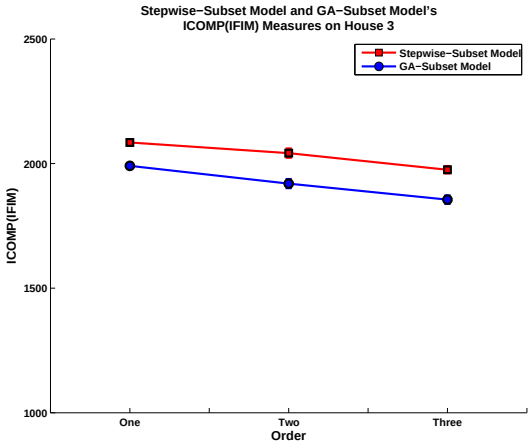
Using the data from the same house, except missing values are now set to zero and the number of candidate sensors is 128, Figure 12 compares results for the Genetic Algorithm and Stepwise Selection methods based on the model with the lowest $ICOMP(IFIM)$ variance. Comparing the $ICOMP(IFIM)$ values from the Genetic Algorithm and Stepwise Selection models (Figure 12(b)) against previous $ICOMP(IFIM)$ values (Figure 11(b) and Figure 10(b)), one will see that there is considerable degradation in the Genetic Algorithm's performance, while the Stepwise Selection is showing increases in performance for all orders. However, the model generated by Stepwise Selection for Markov Order 3 has a fairly large standard deviation, implying the model is highly variable and unstable. In addition, one should notice that both methods are more than likely over-fitting or under-fitting the training examples as the Markov Order increases, which is clearly visible from the decreasing performance in the goodness-of-fit (Figure 12(e)). The Genetic Algorithm uses 63 sensors, 93 sensors, and 109 sensors, while Stepwise Selection uses 71 sensors, 91 sensors, and 75 sensors.

Changing the best model selection strategy to selecting the model with the lowest mean $ICOMP(IFIM)$ increases overall performance on the Campbell Creek House 3 dataset with 128 candidate sensors for all models generated by both methods (Figure 13(b)). However, Stepwise Selection has a slightly better goodness-of-fit for orders 2 and 3 compared to the Genetic Algorithm (Figure 13(e)), but Stepwise Selection's model complexity is much higher than the model complexity for the Genetic Algorithm for Order 2. Yet, both methods have equivalent complexity for Markov Order 3, making the Stepwise Selection model the best model compared to the previous models in Figure 12(e), Figure 10(e), and Figure 11(e). In addition, the Stepwise Selection method uses 76 sensors, 86 sensors, and 85 sensors, while the Genetic Algorithm method uses 77 sensors, 88 sensors, and 107 sensors. This implies that the model generated from using the Genetic Algorithm in Figure 12 is over-fitting for higher Markov Orders, and the model generated from Stepwise Selection, also shown in Figure 12, is under-fitting for Markov Order 3 and over-fitting for Markov Order 2.

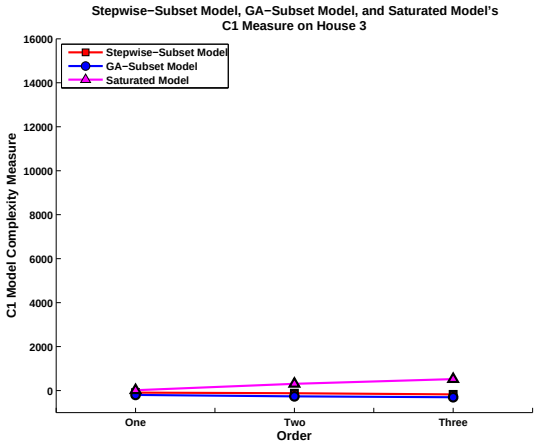
From Figures 10, 11, 12, and 13 it is clear that the best Genetic Algorithm Model and Stepwise Selection Model for House 3 are the models presented in Figure 13. Additionally, we observe, yet again, that dropping variables with missing values had a significant impact on the Stepwise Selection method, and setting missing values to zero showed impact on the Genetic Algorithm method. While the Genetic Algorithm is for the most part producing better models on this data set, Stepwise Selection produced the best model, Markov Order



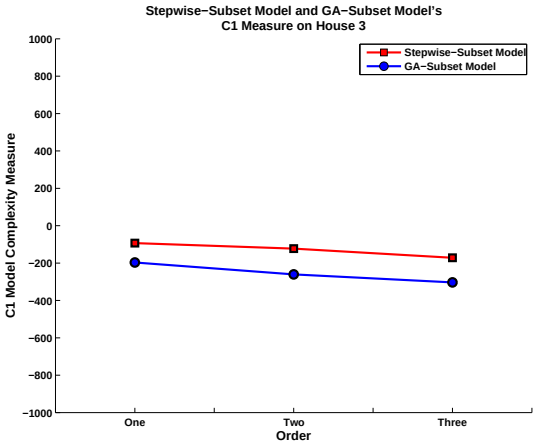
(a) Saturated Model's ICOMP



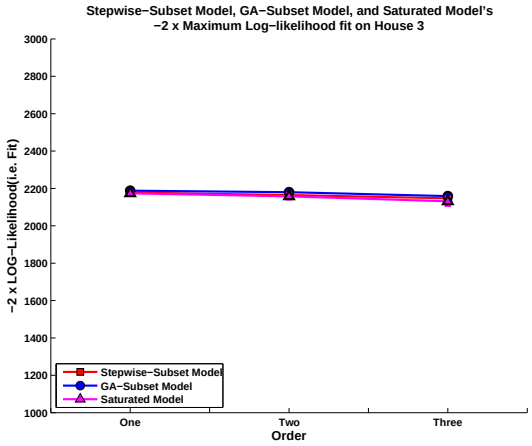
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

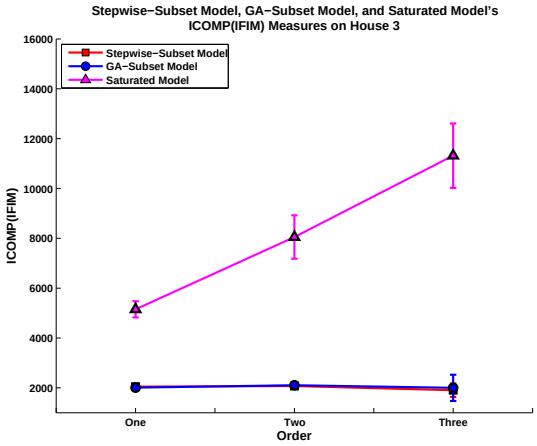


(d) GA and Stepwise Models' Complexity

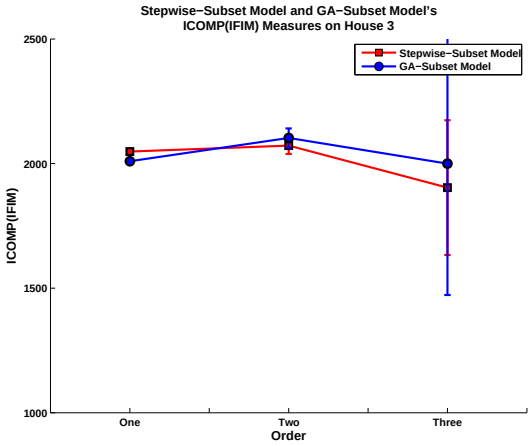


(e) Goodness-of-Fit

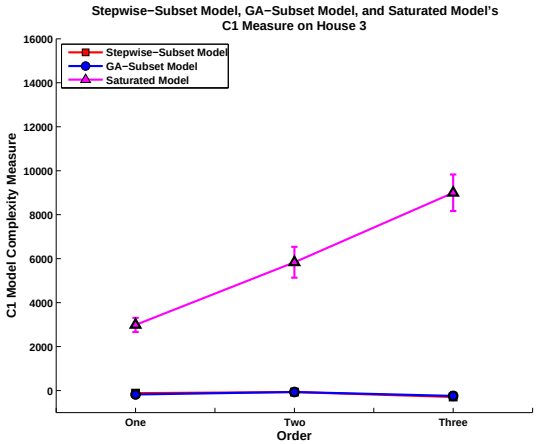
Figure 11: These graphs illustrate the experimental results from applying the models with the lowest mean $ICOMP(IFIM)$ on Campbell Creek House 3. Variables with missing data were removed from the dataset for these results.



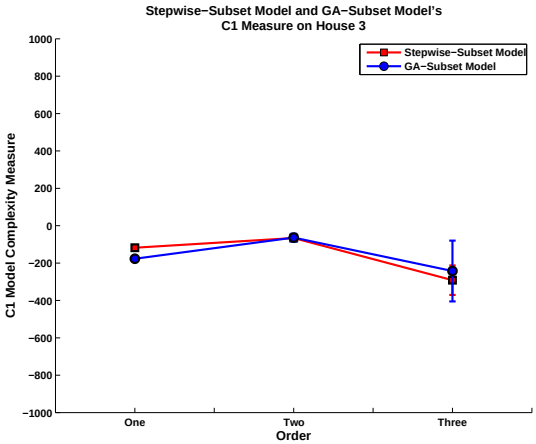
(a) Saturated Model's ICOMP



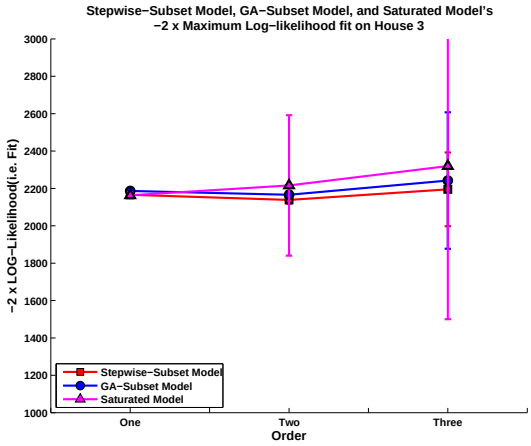
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

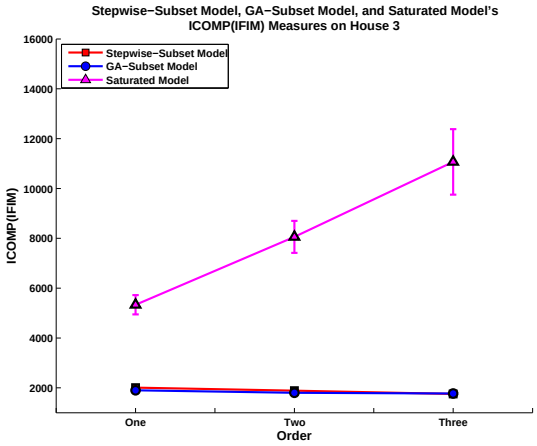


(d) GA and Stepwise Models' Complexity

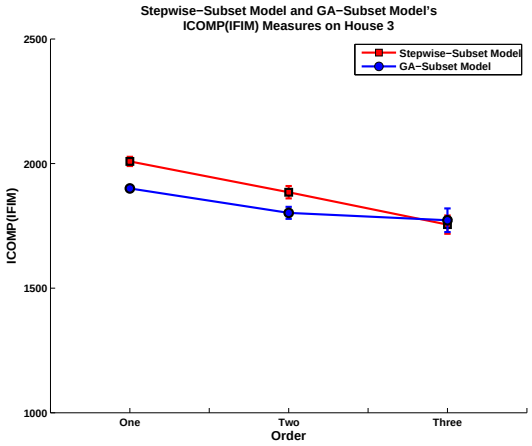


(e) Goodness-of-Fit

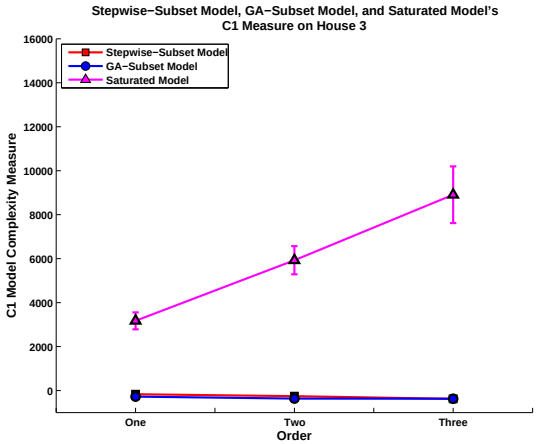
Figure 12: These graphs illustrate the experimental results from applying the models with the lowest $ICOMP(IFIM)$ variance on Campbell Creek House 3. All missing values in the data were set to zero for these results.



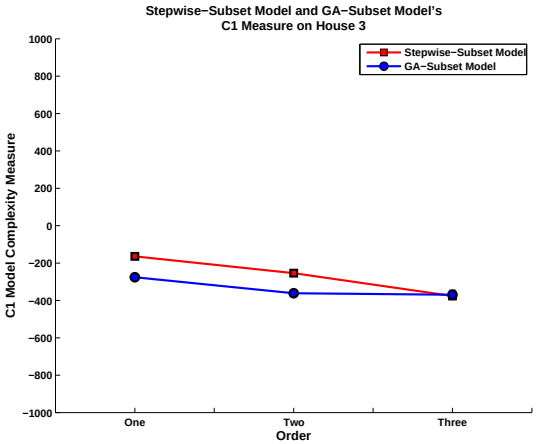
(a) Saturated Model's ICOMP



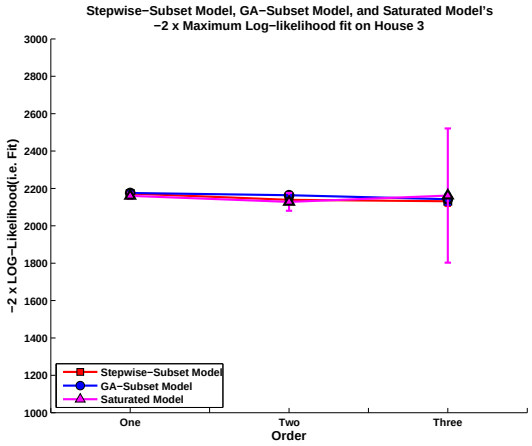
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity



(d) GA and Stepwise Models' Complexity



(e) Goodness-of-Fit

Figure 13: These graphs illustrate the experimental results from applying the models with the lowest mean $ICOMP(IFIM)$ on Campbell Creek House 3. All missing values in the data were set to zero for these results.

3 model in Figure 13, making it the better choice for this particular dataset.

6.4 Across All Houses

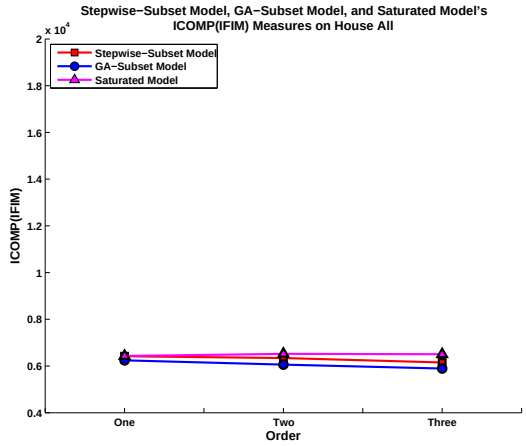
Figure 14 compares the results of the Genetic Algorithm and Stepwise Selection Wrappers when selecting the best model based on lowest $ICOMP(IFIM)$ variance, across all Campbell Creek Houses. In addition, variables that have missing values were dropped, leaving each method with 75 candidate sensors. According to the $ICOMP(IFIM)$ values in Figure 14(a), the Genetic Algorithm is generating better models than Stepwise Selection for all Markov Orders. The goodness-of-fit is equivalent for all models generated with each method (Figure 14(e)), implying that the Genetic Algorithm is consistently minimizing model complexity and maintaining goodness-of-fit. The Genetic Algorithm is using 50 sensors, 61 sensors, and 69 sensors, while Stepwise Selection is using 56 sensors, 63 sensors, and 62 sensors.

Changing the best model selection strategy to one of selecting the model with the lowest mean $ICOMP(IFIM)$ value across all houses, with 75 candidate sensors, one will see that the Genetic Algorithm's $ICOMP(IFIM)$ values in Figure 15(b) indicate no changes in performance quality. However, comparing the Stepwise Selection results in the same Figure to the results in Figure 14(b), one sees a slight increase in performance for Markov Order 2, and the results for Markov Order 1 and 3 are about the same. In addition, the goodness-of-fits for these models (Figure 15(e)) are identical to the goodness-of-fits observed in Figure 14(e). The Genetic Algorithm is using 62 sensors, 62 sensors, and 68 sensors, while Stepwise Selection is using 55 sensors, 61 sensors, and 67 sensors.

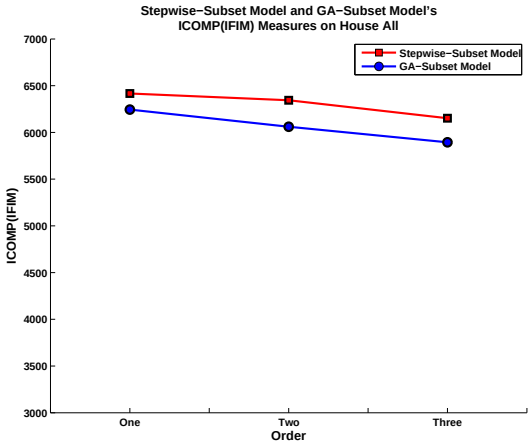
Using the same data, except missing values are now set to zero and the number of candidate sensors is 141, Figure 16 compares results for the Genetic Algorithm and Stepwise Selection methods based on the model with the lowest $ICOMP(IFIM)$ variance. Figure 16(b) shows that the Genetic Algorithm and Stepwise Selection methods' $ICOMP(IFIM)$ values have a very large increase in performance compared to previous results in Figure 14(b) and Figure 15(b). The increase performance mainly stems from both methods showing decreases in model complexity (Figure 16(d)), but there are slight improvements in goodness-of-fit (Figure 16(e)) as well. The Genetic Algorithm uses 93 sensors, 123 sensors, and 118 sensors, while Stepwise Selection uses 98 sensors, 107 sensors, and 109 sensors.

Changing the best model selection strategy to selecting the model with the lowest mean $ICOMP(IFIM)$ increases overall performance across all houses with 141 candidate sensors for all models generated by both methods (Figure 17(b)). Comparing the model complexity for both models in Figure 17(d) with all previous models on this data set, one will see that these models obtain the lowest complexity for Markov Orders 2 and 3, and the same model complexity as the models seen in Figure 16(d), for Markov Order 1. Additionally, the goodness-of-fit (Figure 17(e)) for these models is essentially the same as the goodness-of-fit presented for the models seen in Figure 16(e). This best performing Genetic Algorithm algorithm model uses 95 sensors, 123 sensors, and 124 sensors, while the best performing Stepwise Selection uses 85 sensors, 104 sensors, and 110 sensors.

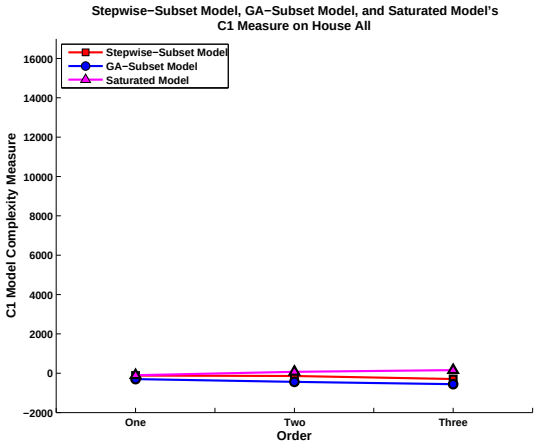
From Figures 14, 15, 16, and 17 it is clear that the best Genetic Algorithm Model and Stepwise Selection Model across all houses are presented in Figure 17. In the previously



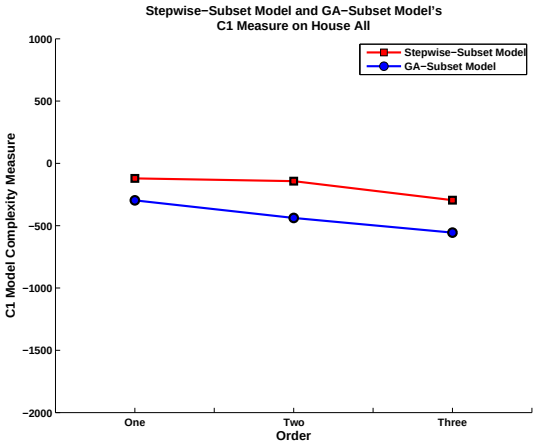
(a) Saturated Model's ICOMP



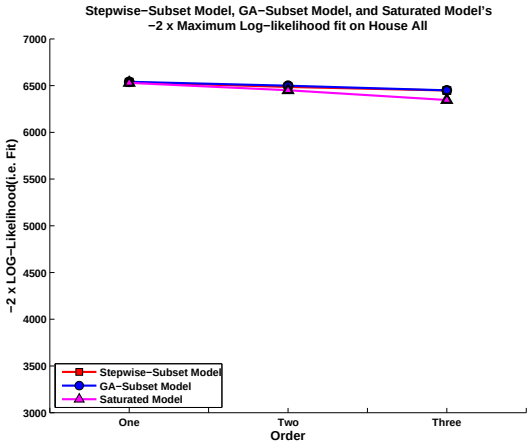
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

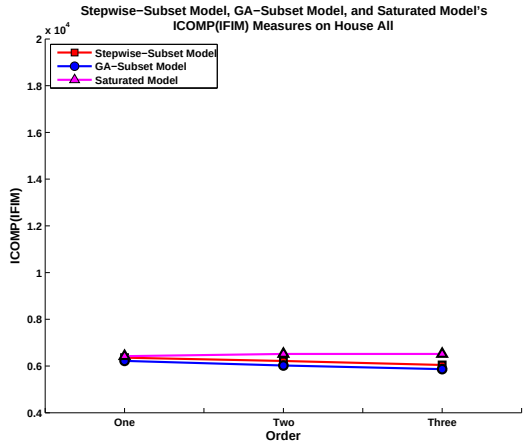


(d) GA and Stepwise Models' Complexity

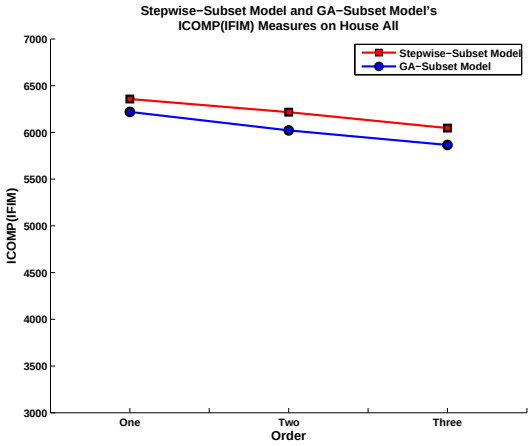


(e) Goodness-of-Fit

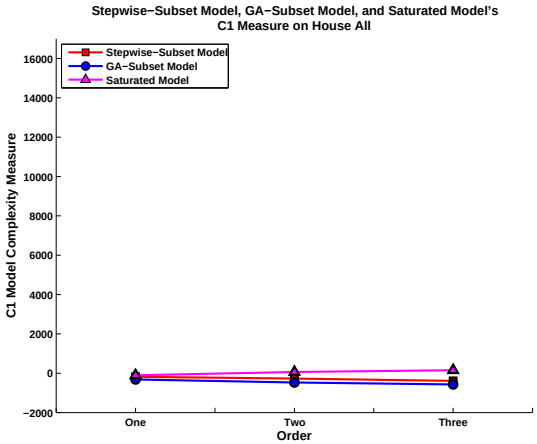
Figure 14: These graphs illustrate the experimental results from applying the models with the lowest $ICOMP(IFIM)$ variance across all houses. Variables with missing data were removed from the dataset for these results.



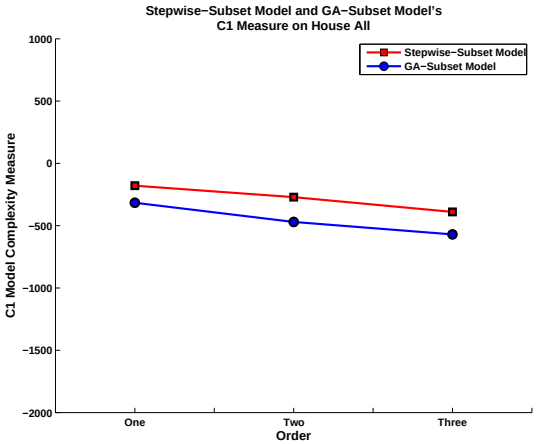
(a) Saturated Model's ICOMP



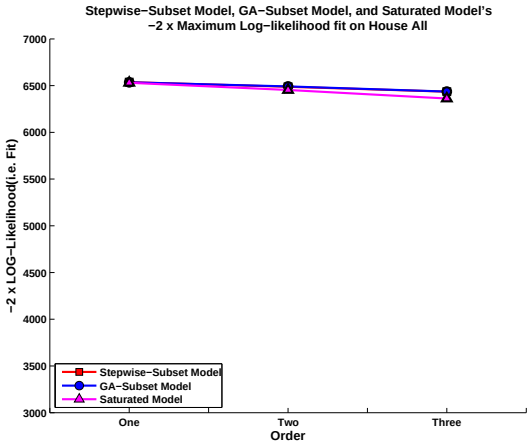
(b) GA and Stepwise Model's ICOMP



(c) Saturated Model's Complexity

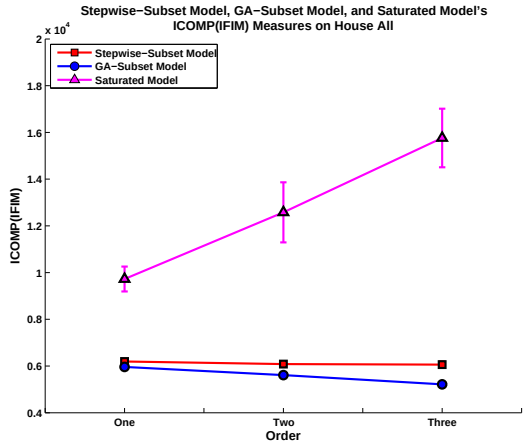


(d) GA and Stepwise Model's Complexity

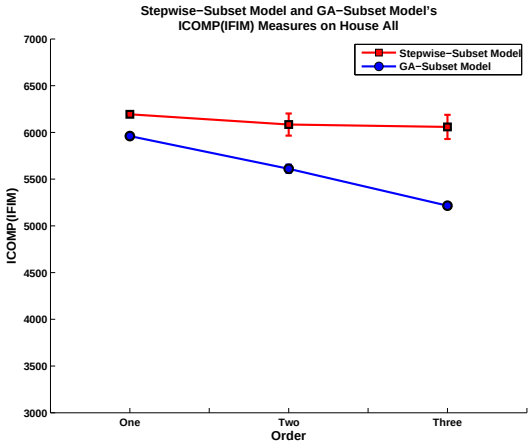


(e) Goodness-of-Fit

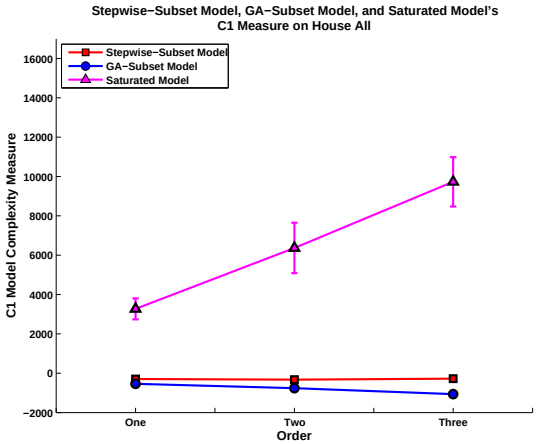
Figure 15: These graphs illustrate the experimental results from applying the models with the lowest mean $ICOMP(IFIM)$ across all houses. Variables with missing data were removed from the dataset for these results.



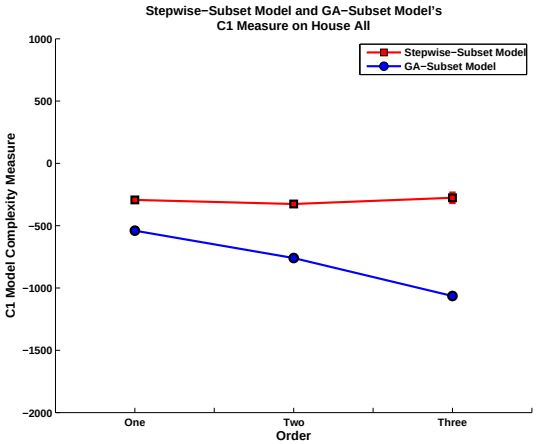
(a) Saturated Model's ICOMP



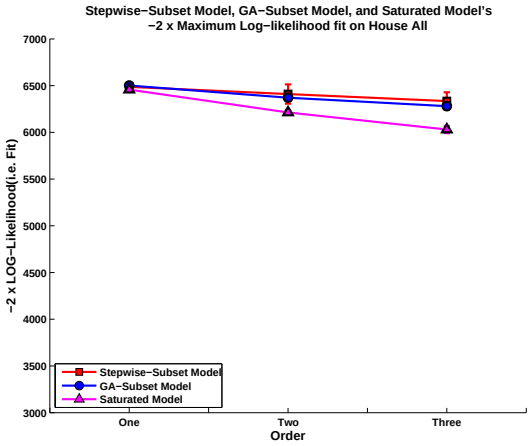
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

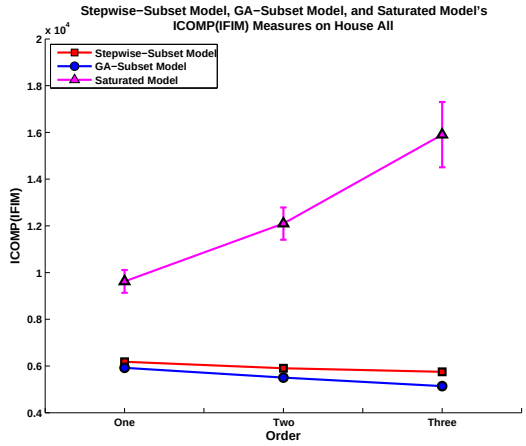


(d) GA and Stepwise Models' Complexity

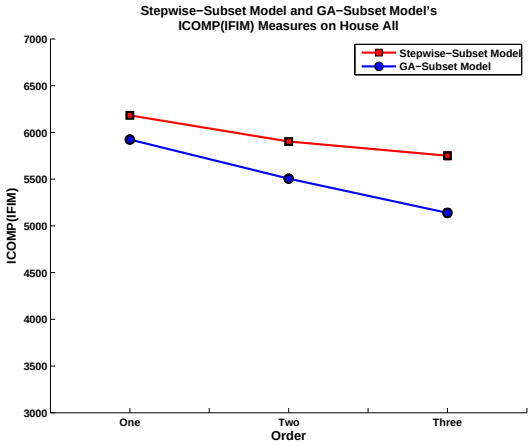


(e) Goodness-of-Fit

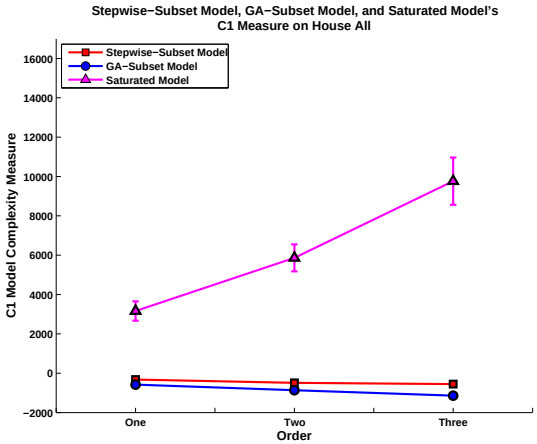
Figure 16: These graphs illustrate the experimental results from applying the models with the lowest $ICOMP(IFIM)$ variance across all houses. All missing values in the data were set to zero for these results.



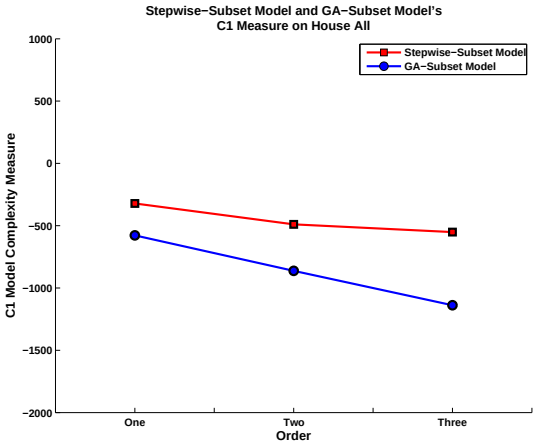
(a) Saturated Model's ICOMP



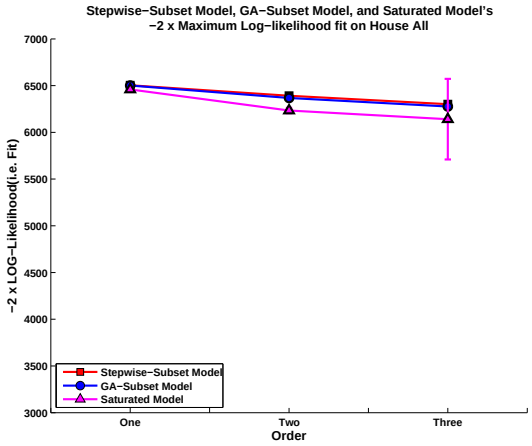
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity



(d) GA and Stepwise Models' Complexity



(e) Goodness-of-Fit

Figure 17: These graphs illustrate the experimental results from applying the models with the lowest mean $ICOMP(IFIM)$ across all houses. All missing values in the data were set to zero for these results.

presented results, Stepwise Selection was generally the only method significantly affected by dropping variables with missing values; however, the Genetic Algorithm method was greatly affected as well on this data set. The key reason for this change is due to the fact that not all the houses have the same sensors, and dropping sensors with missing values greatly limits the number available sensors, which greatly restricts the Genetic Algorithm's search space.

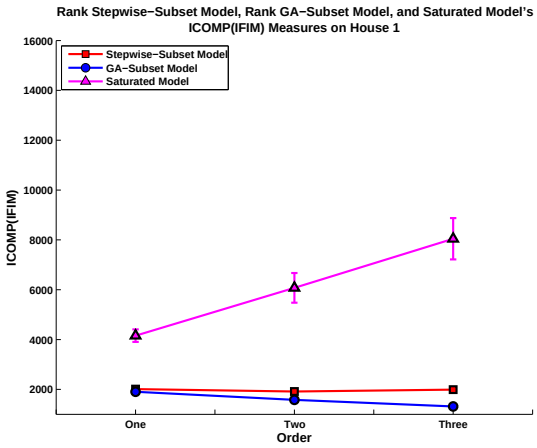
6.5 Variable Ranking

Figures 18, 19, 20, and 21 present the results from applying our sensor ranking technique to determine the best model, when variables with missing values were removed. Recall that the sensor ranking method combines all best models found for each method, and then selects the top k sensors from the list to use in the final model. For all of these results, we heuristically set k equal to the number of sensors whose total vote is greater than zero. Additionally, Tables 5 and 6 show the top ten sensors for both methods on Markov Order 1.

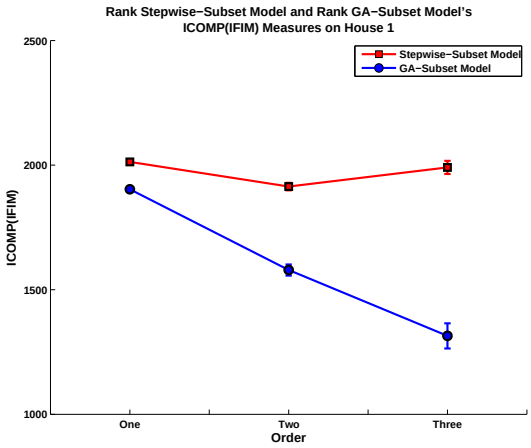
Comparing the results from Figure 18 with the previous results for Campbell Creek House 1 (Figures 2, 3, 4, and 5), one can see that the Rank Model created from combining all the models generated by the Genetic Algorithm is better than the previously seen best models on House 1 (Figure 5). However, the Rank model constructed from the Stepwise Selection models in Figure 18 is worse than the previously seen best Stepwise Selection model on House 1 (Figure 5), but is better than the model seen in Figure 2 for all Markov Orders, and is better than the model in Figure 4 for Markov Order 1 and 2. Combining Stepwise models, where variables with missing values were removed, gives some improvement in performance, but most likely the removed variables are contributing to the poor performance. This Stepwise Rank Model is created using models that have previously demonstrated poor performance, because the variables with missing data were removed. This means one cannot expect a large performance increase when the base models are poor.

On Campbell Creek House 2, the Rank Model created from combining the Genetic Algorithm subset models compared to all previously presented models (Figures 6, 7, 8, and 9) is the worst model (Figure 19). The model's complexity is fairly close to the fully Saturated Model (Figure 23(c)) for all Markov Orders, making it much more undesirable than the previously presented models. The Rank Model constructed from the Stepwise Selection models performs much better than the Rank Genetic Algorithm model, but is not better than the best model seen in the previously presented House 2 results. The Stepwise Rank Model's Markov Order 3 (Figure 19(a)) has better performance than the Stepwise Selection model for the Markov Order 3 seen in Figure 7(a), but worse performance on Markov Order 1 and 2. Additionally, comparing the same rank model to the Stepwise Selection results in Figure 6(a), we observe that Rank Model's Markov Order 1 performance is worse, but the performance is better for the other Markov Orders. Comparing the Stepwise Rank Model's result against the Stepwise selected models without removed variables, we see that previous results in Figure 9 are better for all Markov Orders, and the results in Figure 8 are better for Markov Order 1 and 2.

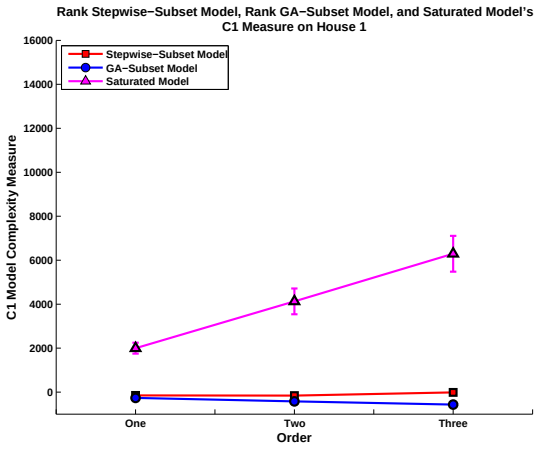
Comparing the Rank Model created from the Genetic Algorithm for House 3 (Figure 20), where variables with missing values were dropped, against all previously presented results



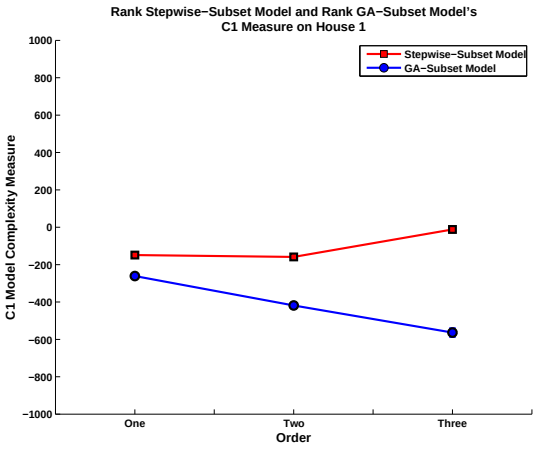
(a) Saturated Model's ICOMP



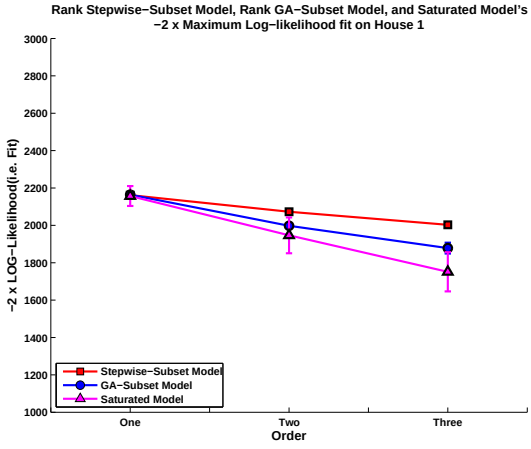
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

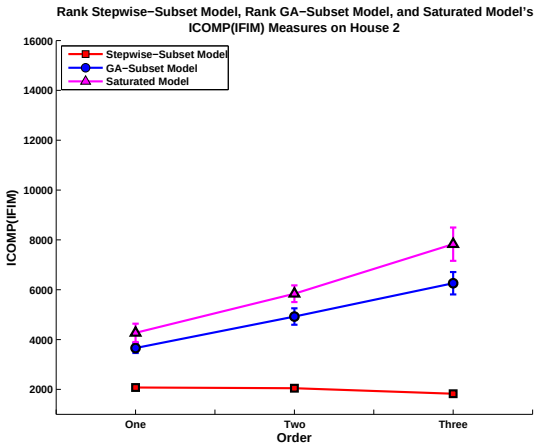


(d) GA and Stepwise Models' Complexity

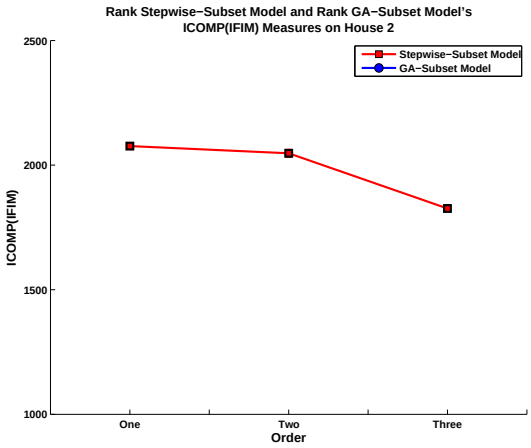


(e) Goodness-of-Fit

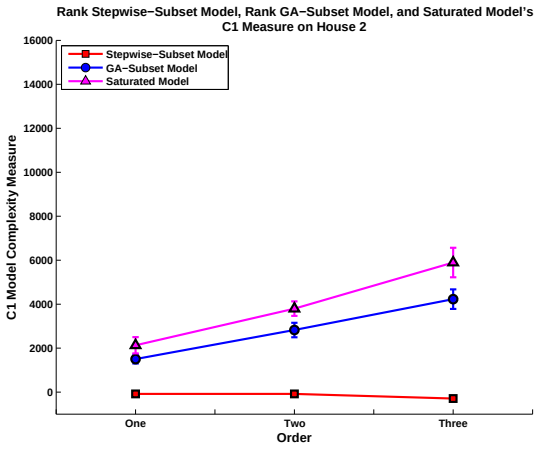
Figure 18: Experimental results for Campbell Creek House 1's Rank Models with dropped variables that have missing data.



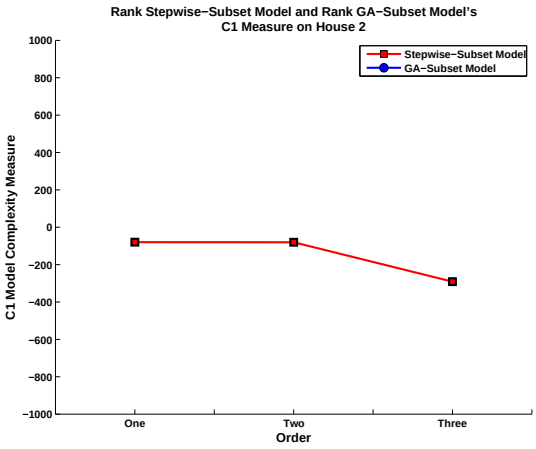
(a) Saturated and GA Models' ICOMP



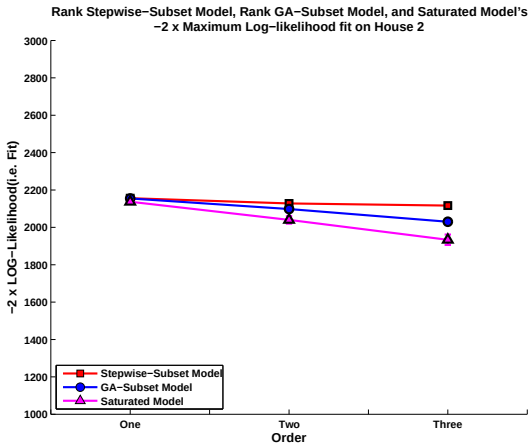
(b) Stepwise Model's ICOMP



(c) Saturated and GA Models' Complexity



(d) Stepwise Model Complexity



(e) Goodness-of-Fit

Figure 19: Experimental results for Campbell Creek House 2's Rank Models with dropped variables that have missing data.

House 1	House 2	House 3	Across All
HW Tot	HW Tot	HP1 in Tot	HP1 in Tot
bathup lts Tot	bathup lts Tot	HP1 out Tot	HP1 out Tot
LVL1 lts Tot	LVL1 lts Tot	HP1 back Tot	HP1 in fan Tot
Kit tmp Avg	wash Tot	HP1 comp Tot	HP2 in Tot
BedB tmp Avg	LVL1 plg Tot	FanTech Tot	HP2 out Tot
Nrake1 tmp Avg	RoofN tmp Avg	solar HW pump Tot	FanTech Tot
Nrake2 tmp Avg	AtticN tmp Avg	bathup lts Tot	solar HW pump Tot
Srake1 tmp Avg	WallNcav tmp Avg	LVL1 lts Tot	HW Tot
Attic tmp Avg	BedM tmp Avg	bed Tot	bathup lts Tot
WashHot flow Tot	Bed2 tmp Avg	dryer Tot	LVL1 lts Tot

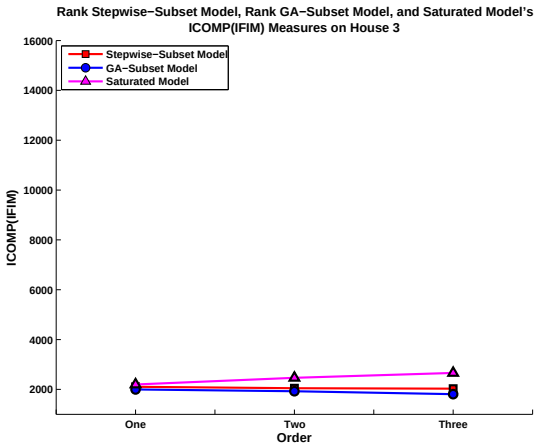
Table 5: Top 10 Sensors from the Voted Markov Order 1 models per house. The Markov Order 1 models were constructed by combining all the best Stepwise Selection subsets, using the voting process discussed in Section 3.14. Additionally, the best Stepwise Selection subsets were computed using datasets where variables with missing values were removed.

for House 3 (Figures 10, 11, 12, and 13), one can see that the rank model for Markov Order 2 and Markov Order 3 has better performance than the Genetic Algorithm results in Figure 11, and Markov Order 1 performance is the same. In addition, the Genetic Algorithm results in Figure 10 are worse than the Genetic Rank Model for Markov Orders 2 and 3, but the same for Markov Order 1. Comparing the Genetic Rank Model results against the Genetic Algorithm results in Figure 13, one will see that the Genetic Rank Model is worse for all Markov Orders. Yet, the Genetic Rank Model produces better results than the Genetic Algorithm results presented in Figure 12. The Stepwise Rank Model results are generally similar to the previous Stepwise Selection results on House 3, and only present better performance when compared against the poorer performing Stepwise models.

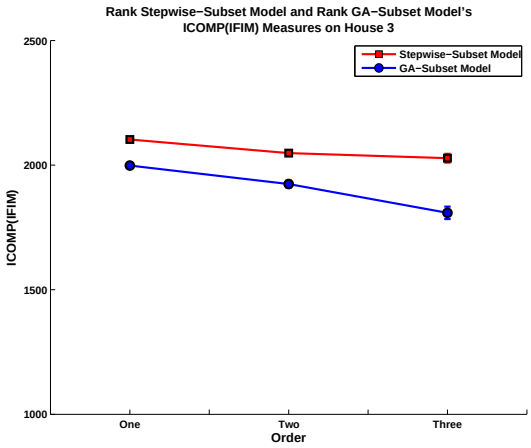
Lastly, the rank models created from the Genetic Algorithm and Stepwise Selection models across all houses perform the same as the results presented in Figure 10 and Figure 11. This shows that the ranking process is not degrading performance across all the houses, but it is not improving performance like it has for certain models on House 1 and House 2.

Figures 22, 23, 24, and 25 present the results from applying our sensor ranking technique to determine the best model, when missing values are set to zero. In addition, Tables 7 and 8 show the top ten sensors for both methods on Markov Order 1. All the Stepwise rank models shown in these figures are extremely similar to the previous Stepwise rank models (Figures 18, 19, 20, and 21) and do not provide performance increase, but do not decrease performance drastically either. In addition, the Genetic Rank Model on House 2 performs poorly in terms of model complexity, just like the Genetic Rank Model seen in Figure 19.

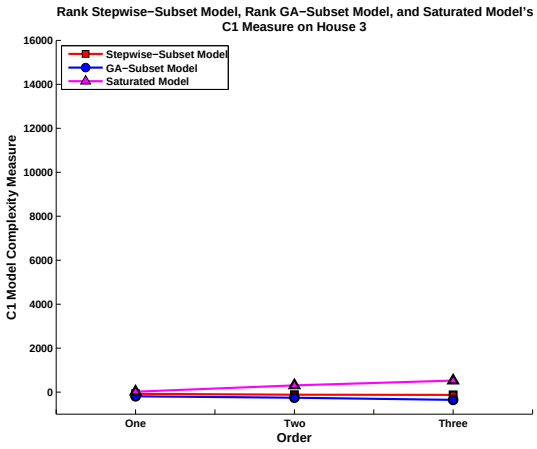
The key observation is that the Genetic Rank models in Figures 22, 24, and 25 present the best model for House 1, House 3, and across all houses. On House 1, the Markov Order 3 model provides the best goodness-of-fit compared to all previous results and has better model complexity than all previous models. The Genetic Rank model across all houses has the best goodness-of-fit and best complexity compared to all previous models as well. Lastly,



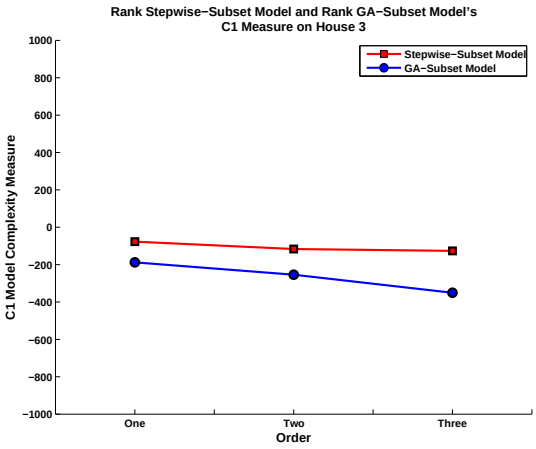
(a) Saturated Model's ICOMP



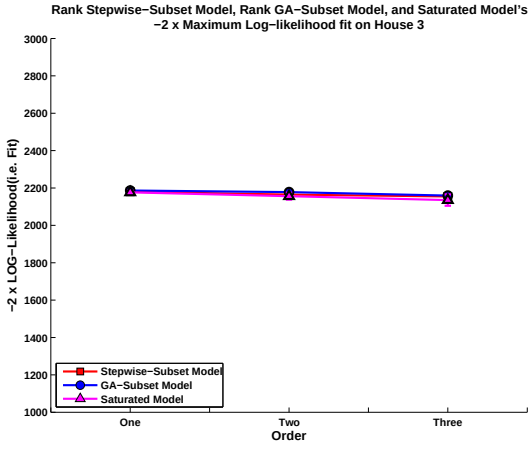
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

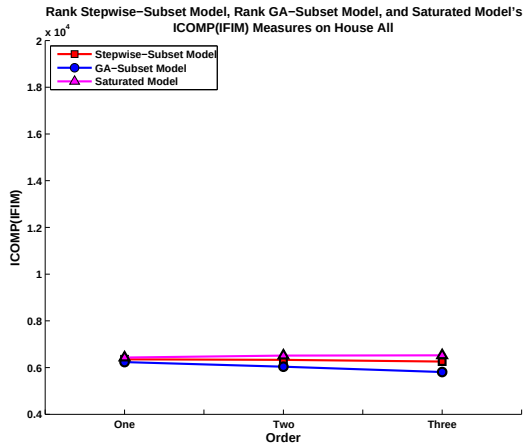


(d) GA and Stepwise Models' Complexity

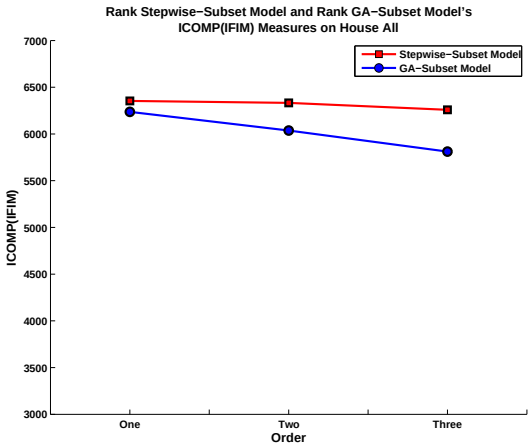


(e) Goodness-of-Fit

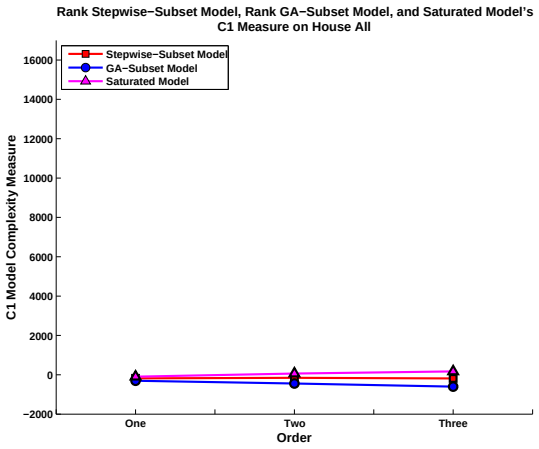
Figure 20: Experimental results for Campbell Creek House 3's Rank Models with dropped variables that having missing data.



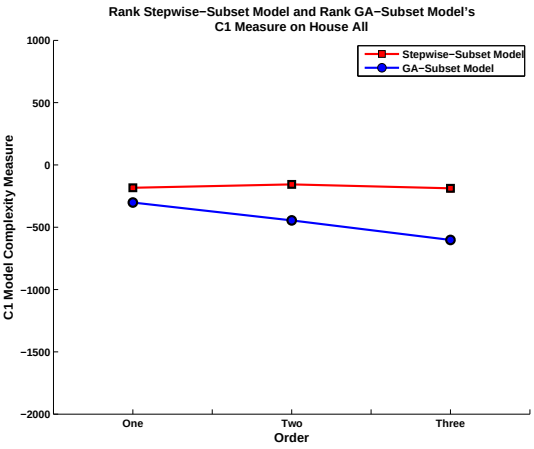
(a) Saturated Model's ICOMP



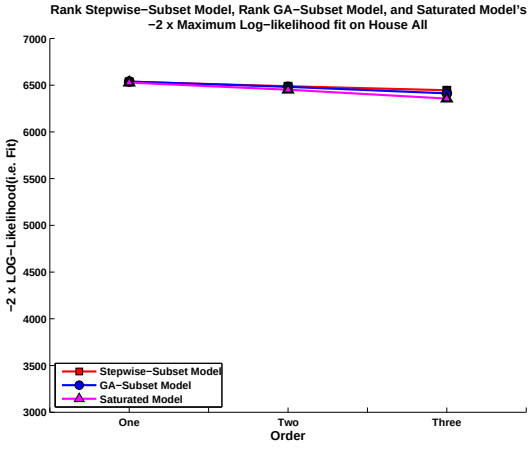
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity



(d) GA and Stepwise Models' Complexity



(e) Goodness-of-Fit

Figure 21: Experimental results for Rank Models across all houses with dropped variables that have missing data.

House 1	House 2	House 3	Across All
gar ext lts Tot	LVL1 lts Tot	bathup lts Tot	HP1 out Tot
LVL1 lts Tot	gar ext plg Tot	LVL1 lts Tot	FanTech Tot
bath plg Tot	CantFlr RH Avg	dryer Tot	LVL1 lts Tot
gar ext plg Tot	AtticN HFT Avg	wash Tot	bed Tot
bed Tot	HP1ret tmp Avg	micro Tot	dryer Tot
dish Tot	Attic RH Avg	range Tot	wash Tot
RoofS HFT Avg	FreshAir Flow Tot	FanTexh RH Avg	LVL1 plg Tot
fridge Tot	gar ext lts Tot	HW Tot	gar ext plg Tot
CondenHP1 Tot	CondenHP1 Tot	WallScav RH Avg	micro Tot
HP2sup RH Avg	WallScav RH Avg	FanTech ToT	dish Tot

Table 6: Top 10 Sensors from the Voted Markov Order 1 models per house. The Markov Order 1 models were constructed by combining all the best Genetic Algorithm subsets, using the voting process introduced in Section 3.14. Additionally, the best Genetic Algorithm subsets were computed using datasets where variables with missing values were removed.

the Genetic Rank Model improves model complexity greatly on House 3, making it the best performing model in terms of $ICOMP(IFIM)$ for all Markov Orders.

6.6 Ground Truth Comparison

An advantage of our model selection approach is that it can allow a practical search over a large solution space to find good solutions that work well in practice. Comparing it to the "Ground Truth" solution is computationally infeasible. Nevertheless, it is informative to calculate the exact solution for small problems, in order to provide comparative results to our approach. We, therefore, calculated the best sensors subsets, "Restricted Ground Truth," with cardinality up to four. We refer to these sensor subsets as "Restrict Ground" because they are globally optimal solutions to a smaller problem. Tables 9 and 10 show the "Restricted Ground Truth" subsets for datasets with removed missing data variables, while Tables 11 and 12 show the "Restricted Ground Truth" subsets for datasets with missing data values set to zero. These subsets were computed in a brute force fashion by selecting the best subset from all possible subsets that minimized the linear regression's residual SSE (Sum Squared Error). These tables compare the "Restricted Ground Truth's" Coefficient of Variance (CV) and ICOMP scores against the best found Genetic Algorithm Models with Markov Order 1 and the best Top 10 Sensor lists on each respective dataset. This means that we are only comparing "Restricted Ground Truth" results for a dataset against models that were found using the same dataset. The "Top 10 Sensors" results in Tables 9 and 10 were generated using the sensor listings found in Table 6 for House 1, 3, and across all, while the results for House 2 were generated using the sensor listings found in Table 5. The "Top 10 Sensors" results in Tables 11 and 12 were generated using the sensor listings found in Table 8.

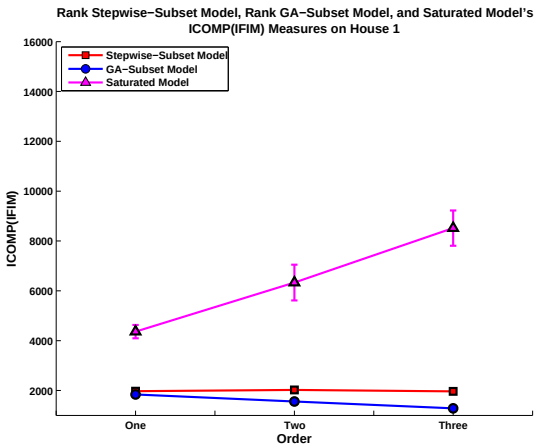
Analyzing Tables 9 and 10 shows that there is a large difference in performance between

House 1	House 2	House 3	Across All
bathup lts Tot	bathup lts Tot	HP1 in Tot	bathup lts Tot
bed Tot	LVL1 lts Tot	HP1 out Tot	LVL1 lts Tot
dish Tot	wash Tot	HP1 back Tot	wash Tot
range Tot	LVL1 plg Tot	HP1 comp Tot	micro Tot
WallScav tmp Avg	RoofN tmp Avg	bathup lts Tot	dish Tot
BedM tmp Avg	AtticN tmp Avg	LVL1 lts Tot	CantFlr tmp Avg
Bed3 tmp Avg	WallNcav tmp Avg	bed Tot	BedM tmp Avg
BedB tmp Avg	BedM tmp Avg	wash Tot	Bed3 tmp Avg
Nrake1 tmp Avg	Bed2 tmp Avg	micro Tot	Bed2 tmp Avg
Nrake2 tmp Avg	Mbath tmp Avg	RoofS tmp Avg	BedB tmp Avg

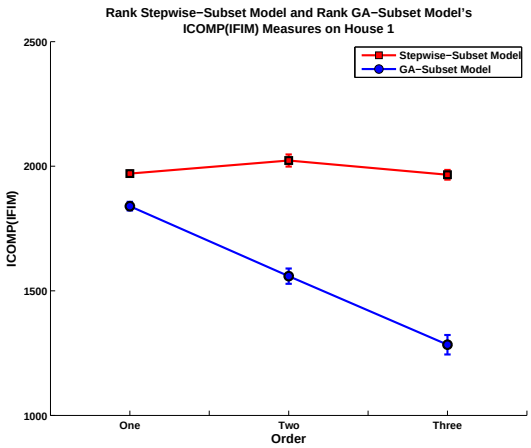
Table 7: Top 10 Sensors from the Voted Markov Order 1 models per house. The Markov Order 1 models were constructed by combining all the best Stepwise Selection subsets, using the voting process discussed in Section 3.14. Additionally, the best Stepwise Selection subsets were computed with missing data values set to zero.

House 1	House 2	House 3	Across All
HWcold tmp Avg	wash Tot	wash Tot	HWhot tmp Avg
dish Tot	HWcold tmp Avg	HP1 out Tot	washHot tmp Avg
LVL1 lts Tot	gar ext lts Tot	WashHot flow Tot	HP1ret RH Avg
HWhot tmp Avg	gar ext plg Tot	SlrW1 Avg	Nrake5 tmp Avg
TrueNetEnergy	bed Tot	gar ext lts Tot	dishHot tmp Avg
BedB tmp Avg	CantFlr RH Avg	HP1 comp Tot	WallScav RH Avg
HP2sup tmp Avg	fridge Tot	HWHXtoTank tmp Avg	LVL1 lts Tot
RoofS HFT Avg	Nrake5 tmp Avg	AtticFlrS HFT Avg	HWcold tmp Avg
bathup lts Tot	Shower tmp Avg	dryer Tot	CondenHWHP Tot
gar ext lts Tot	WallScav RH Avg	bed Tot	HP1 in fan Tot

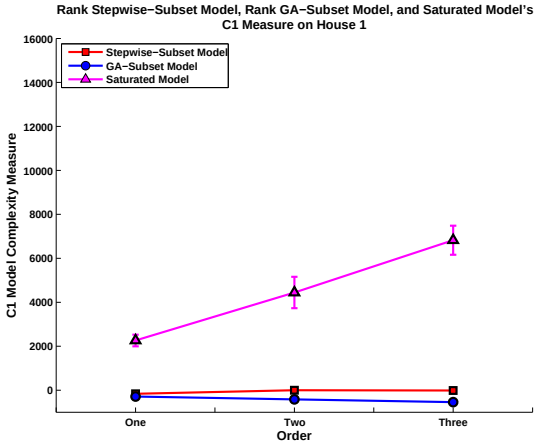
Table 8: Top 10 Sensors from the Voted Markov Order 1 models per house. The Markov Order 1 models were constructed by combining all the best Genetic Algorithm subsets, using the voting process introduced in Section 3.14. Additionally, the best Genetic Algorithm subsets were computed with missing data values set to zero.



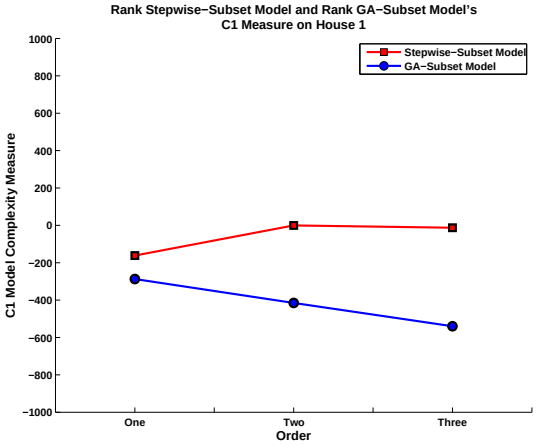
(a) Saturated Model's ICOMP



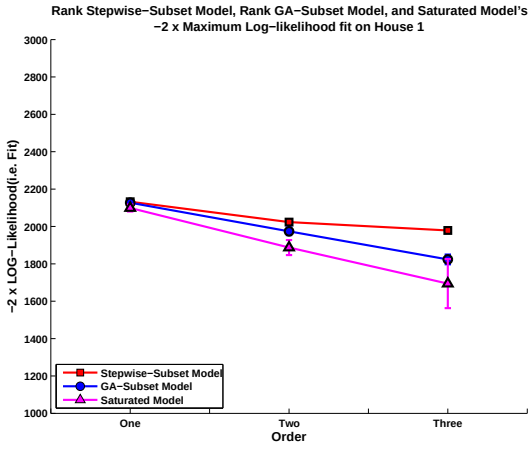
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

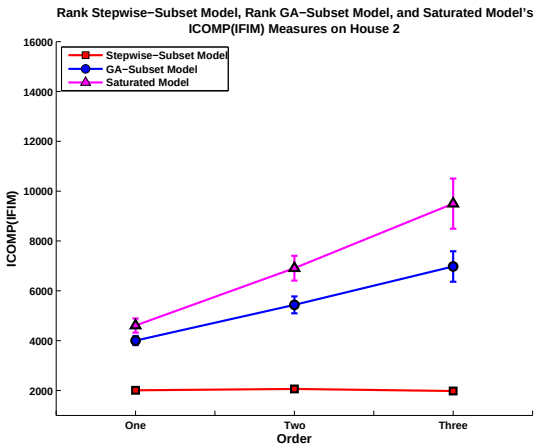


(d) GA and Stepwise Models' Complexity

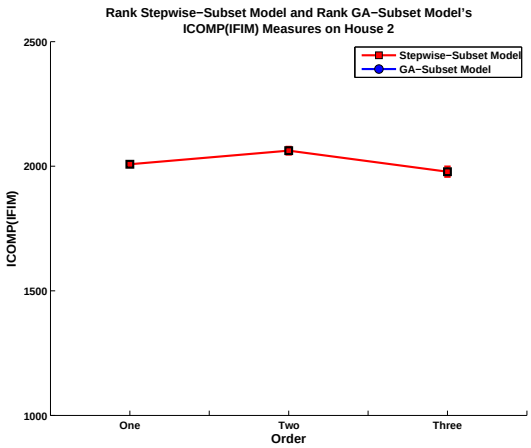


(e) Goodness-of-Fit

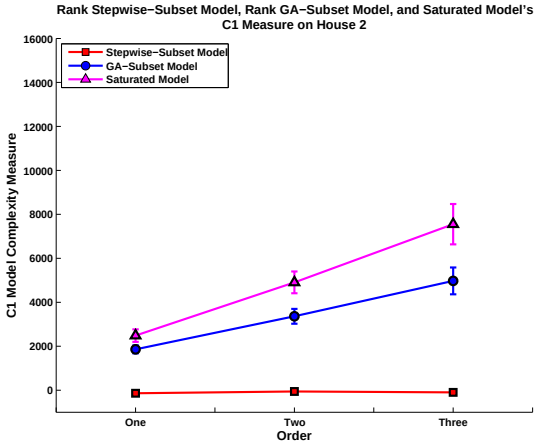
Figure 22: Experimental results for Campbell Creek House 1's Rank Models with missing data values set to zero.



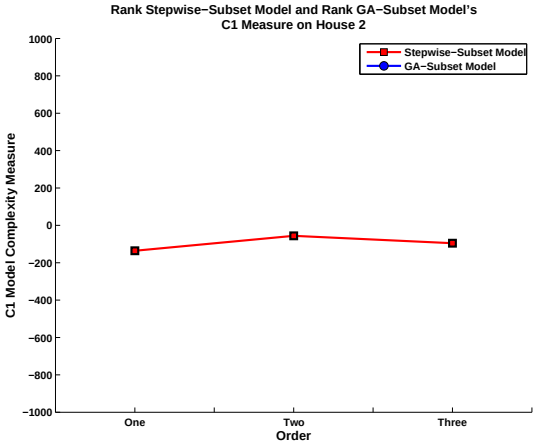
(a) Saturated and GA Models' ICOMP



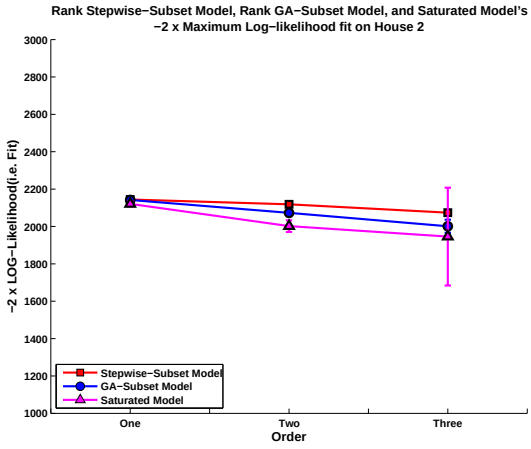
(b) Stepwise Model's ICOMP



(c) Saturated and GA Models' Complexity

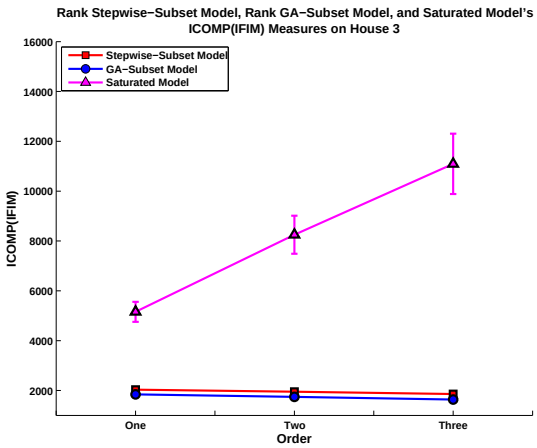


(d) Stepwise Model's Complexity

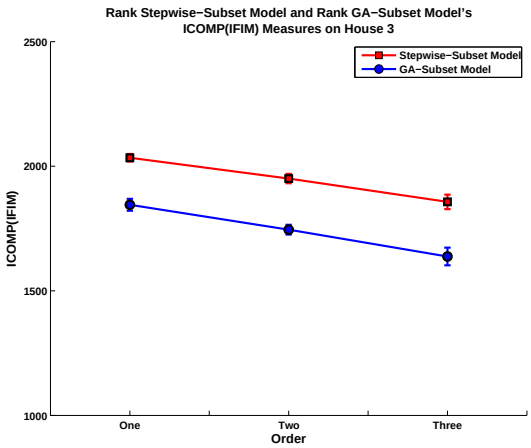


(e) Goodness-of-Fit

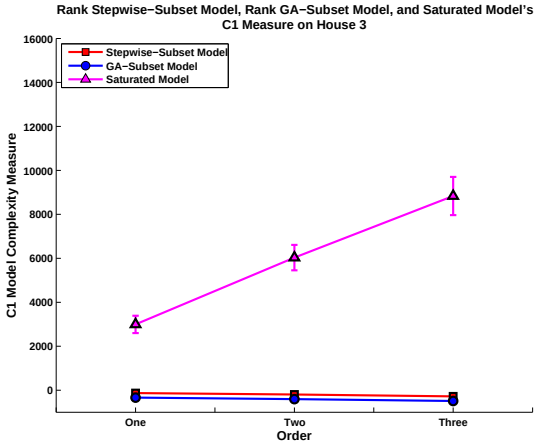
Figure 23: Experimental results for Campbell Creek House 2's Rank Models with missing data values set to zero.



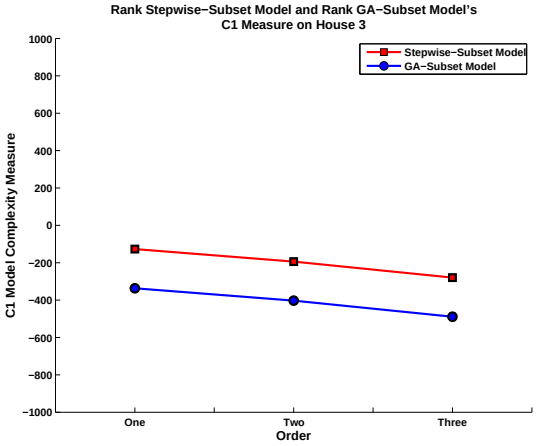
(a) Saturated Model's ICOMP



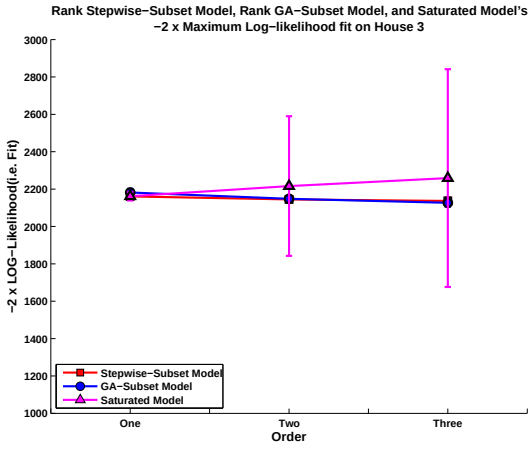
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity

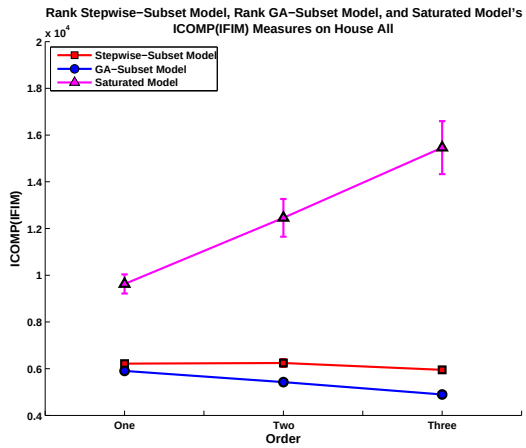


(d) GA and Stepwise Models' Complexity

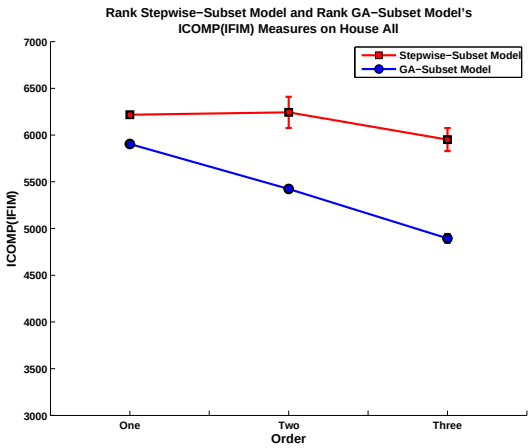


(e) Goodness-of-Fit

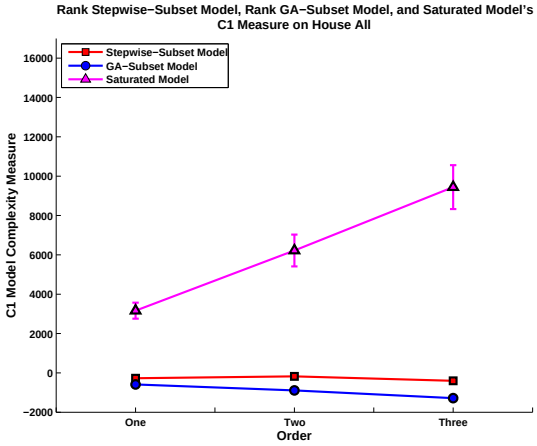
Figure 24: Experimental results for Campbell Creek House 3's Rank Models with missing data values set to zero.



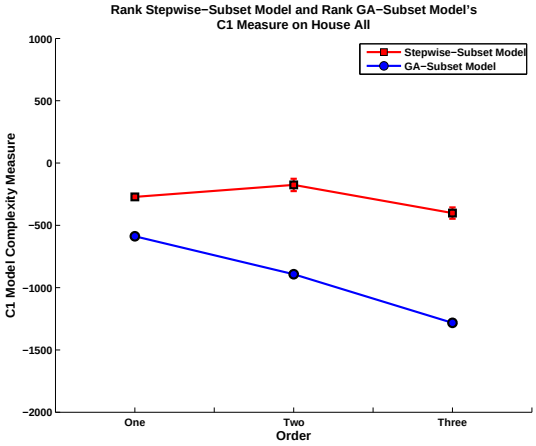
(a) Saturated Model's ICOMP



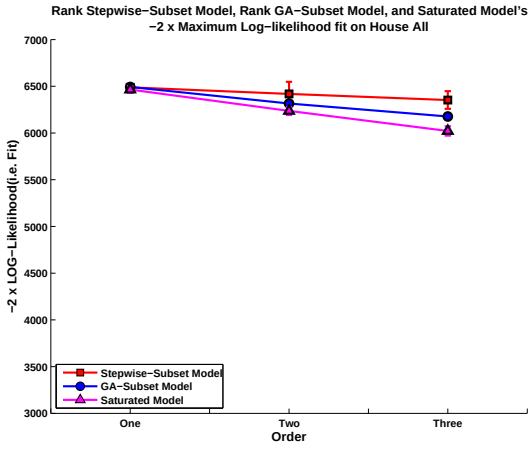
(b) GA and Stepwise Models' ICOMP



(c) Saturated Model's Complexity



(d) GA and Stepwise Models' Complexity



(e) Goodness-of-Fit

Figure 25: Experimental results for applying Rank Models across all houses with missing data values set to zero.

Houses 1

Sensors	Ground Truth CV	Best Model CV	Top 10 Sensors' CV	Ground Truth ICOMP	Best Model ICOMP	Top 10 Sensors' ICOMP
TrueNetEnergy	48.88±0.82	37.26±0.79	57.38±0.87	2190.18±1.40	1902.85±9.55	2139.34±2.34
HW Tot TrueNetEnergy	42.91±0.81	37.26±0.79	57.38±0.87	2183.03±2.32	1902.85±9.55	2139.34±2.34
HW Tot dryer Tot TrueNetEnergy	41.79±0.78	37.26±0.79	57.38±0.87	2175.27±2.86	1902.85±9.55	2139.34±2.34
HW Tot LVL1 lts Tot dryer Tot TrueNetEnergy	41.38±0.82	37.26±0.79	57.38±0.87	2167.20±2.94	1902.85±9.55	2139.34±2.34

Houses 2

Sensors	Ground Truth CV	Best Model CV	Top 10 Sensors' CV	Ground Truth ICOMP	Best Model ICOMP	Top 10 Sensors' ICOMP
TrueNetEnergy	47.35±0.95	37.41±0.82	52.57±0.99	2190.10±2.77	1886.09±12.32	2151.20±2.31
HP1 comp Tot TrueNetEnergy	45.48±0.80	37.41±0.82	52.57±0.99	2184.52±3.42	1886.09±12.32	2151.20±2.31
HP1 comp Tot HP1ret RH Avg TrueNetEnergy	44.68±0.77	37.41±0.82	52.57±0.99	2177.17±3.69	1886.09±12.32	2151.20±2.31
HP1 back Tot HP1 comp Tot LVL1 lts Tot wash Tot	43.22±0.66	37.41±0.82	52.57±0.99	2167.64±2.15	1886.09±12.32	2151.20±2.31

Table 9: The House 1 Table compares House 1's "Restricted Ground Truth" subsets against the best Markov Order 1 Rank Model seen in Figure 18 and against the Top 10 Sensor list for House 1 in Table 6. The House 2 Table compares House 2's "Restricted Ground Truth" subsets against the best Markov Order 1 Genetic Algorithm Model seen in Figure 7 and the best Top 10 Sensor list for House 2 (Table 5). Variables with missing data were removed for these comparisons. The values given are Coefficient of Variance(CV) and ICOMP.

Houses 3

Sensors	Ground Truth CV	Best Model CV	Top 10 Sensors' CV	Ground Truth ICOMP	Best Model ICOMP	Top 10 Sensors' ICOMP
HP1 in fan Tot	56.65±0.81	43.33±0.75	48.16±0.79	2178.24±1.89	1988.68±7.82	2171.15±2.96
HP1 in fan Tot None Tot(8)	53.86±0.78	43.33±0.75	48.16±0.79	2172.06±1.46	1988.68±7.82	2171.15±2.96
HP1 in fan Tot wash Tot None Tot(8)	50.13±0.78	43.33±0.75	48.16±0.79	2164.38±0.79	1988.68±7.82	2171.15±2.96
HP1 in fan Tot wash Tot None Tot(8) Kit tmp Avg	48.96±0.84	43.33±0.75	48.16±0.79	2159.40±0.86	1988.68±7.82	2171.15±2.96

Houses All

Sensors	Ground Truth CV	Best Model CV	Top 10 Sensors' CV	Ground Truth ICOMP	Best Model ICOMP	Top 10 Sensors' ICOMP
HP2 in Tot	62.67±1.06	42.40±0.40	46.76±0.49	6558.50±1.66	6221.73±8.22	6488.53±4.95
HP1 in Tot HP2 out Tot	53.35±0.45	42.40±0.40	46.76±0.49	6549.78±0.87	6221.73±8.22	6488.53±4.95
HP1 in Tot HP1 out Tot HP2 in fan Tot	50.42±0.40	42.40±0.40	46.76±0.49	6541.64±0.93	6221.73±8.22	6488.53±4.95
HP1 in Tot HP1 out Tot HP2 in fan Tot LVL1 lts Tot	47.79±0.39	42.40±0.40	46.76±0.49	6532.52±1.30	6221.73±8.22	6488.53±4.95

Table 10: The House 3 Table compares House 1's "Restricted Ground Truth" subsets against the best Markov Order 1 Rank Model seen in Figure 20 and against the Top 10 Sensor list for House 3 in Table 6. The Across All Table compares the "Restricted Ground Truth" subsets across all houses against the best Markov Order 1 Rank Model seen in Figure 21 and the best Top 10 Sensor list across all houses (Table 6). Variables with missing data were removed for these comparisons. The values given are Coefficient of Variance(CV) and ICOMP.

Houses 1

Sensors	Ground Truth CV	Best Model CV	Top 10 Sensors' CV	Ground Truth ICOMP	Best Model ICOMP	Top 10 Sensors' ICOMP
NetEnergy	48.89±0.80	33.31±1.04	40.18±0.82	2189.66±1.12	1843.21±25.73	2126.78±2.39
HW Tot NetEnergy	42.91±0.54	33.31±1.04	40.18±0.82	2182.11±1.16	1843.21±25.73	2126.78±2.39
HW Tot dryer Tot NetEnergy	41.79±0.66	33.31±1.04	40.18±0.82	2174.03±0.75	1843.21±25.73	2126.78±2.39
HW Tot dryer Tot washHot tmp Avg NetEnergy	39.81±0.67	33.31±1.04	40.18±0.82	2166.44±1.24	1843.21±25.73	2126.78±2.39

Houses 2

Sensors	Ground Truth CV	Best Model CV	Top 10 Sensors' CV	Ground Truth ICOMP	Best Model ICOMP	Top 10 Sensors' ICOMP
NetEnergy	47.07±1.20	37.32±0.81	50.32±1.34	2189.60±1.79	1821.50±17.78	2158.56±4.13
HP1 comp Tot NetEnergy	45.13±1.17	37.32±0.81	50.32±1.34	2183.42±1.82	1821.50±17.78	2158.56±4.13
HP1 comp Tot HP1ret RH Avg NetEnergy	44.45±1.11	37.32±0.81	50.32±1.34	2176.10±2.25	1821.50±17.78	2158.56±4.13
Nrake2 tmp Avg Nrake4 tmp Avg None Tot(23) NetEnergy	43.42±1.09	37.32±0.81	50.32±1.34	2183.45±2.43	1821.50±17.78	2158.56±4.13

Table 11: The House 1 Table compares House 1's "Restricted Ground Truth" subsets against the best Markov Order 1 Rank Model seen in Figure 22 and against the best Top 10 Sensor list for House 1 (Table 8). The House 2 Table compares House 2's "Restricted Ground Truth" subsets against the best Markov Order 1 Genetic Algorithm Model seen in Figure 9 and the best Top 10 Sensor list for House 2 (Table 7). Missing data values were set to zero for these comparisons. The values given are Coefficient of Variance(CV) and ICOMP.

Houses 3

Sensors	Ground Truth CV	Best Model CV	Top 10 Sensors' CV	Ground Truth ICOMP	Best Model ICOMP	Top 10 Sensors' ICOMP
NetEnergy	51.25±0.83	40.79±0.88	47.51±1.00	2177.34±1.38	1897.62±12.18	2169.82±2.40
None Tot(19) NetEnergy	49.95±0.90	40.79±0.88	47.51±1.00	2171.49±1.38	1897.62±12.18	2169.82±2.40
None Tot(9) None Tot(27) NetEnergy	48.29±0.86	40.79±0.88	47.51±1.00	2164.67±1.50	1897.62±12.18	2169.82±2.40
wash Tot None Tot(9) None Tot(27) NetEnergy	46.78±0.83	40.79±0.88	47.51±1.00	2157.12±1.27	1897.62±12.18	2169.82±2.40

Houses All

Sensors	Ground Truth CV	Best Model CV	Top 10 Sensors' CV	Ground Truth ICOMP	Best Model ICOMP	Top 10 Sensors' ICOMP
NetEnergy	50.78±0.73	39.77±0.73	63.00±0.75	6557.90±2.70	5914.91±10.11	6517.21±1.65
HW Tot NetEnergy	47.49±0.72	39.77±0.73	63.00±0.75	6550.04±2.54	5914.91±10.11	6517.21±1.65
HW Tot dryer Tot NetEnergy	46.42±0.78	39.77±0.73	63.00±0.75	6541.36±2.43	5914.91±10.11	6517.21±1.65
HW Tot dryer Tot None Tot(14) NetEnergy	45.72±0.75	39.77±0.73	63.00±0.75	6532.55±2.83	5914.91±10.11	6517.21±1.65

Table 12: The House 3 Table compares House 1's "Restricted Ground Truth" subsets against the best Markov Order 1 Rank Model seen in Figure 24 and against the best Top 10 Sensor list for House 3 (Table 8). The Across All Table compares the "Restricted Ground Truth" subsets across all houses against the best Markov Order 1 Rank Model seen in Figure 25 and the best Top 10 Sensor list across all houses (Table 8). Missing data values were set to zero for these comparisons. The values given are Coefficient of Variance(CV) and ICOMP.

the "Restricted Ground Truth" and the best Genetic Algorithm models with Markov Order 1, seen in Figures 18, 7, 20, and 21. The Genetic Algorithm Models have much better performance in terms of CV and ICOMP, which implies that the best performing optimal subset is larger than the ones we have computed. However, these best performing approximations use 50 or more sensors. This makes it very difficult to estimate the best performing optimal subset's actual size and to estimate whether one can feasibly compute it directly.

Comparing the same "Restricted Ground Truth" CV and ICOMP results with the best "Top 10 Sensor" lists results shows that the voting scheme is able to produce lower ICOMP values, but overall worse CV results. This implies solving for a small subset directly is better than selecting a small subset using our variable ranking procedure. However, if one is concerned about the best subset being generalizable, then one can solve directly for the best subset using ICOMP as the criteria function rather than CV.

Tables 11 and 12 illustrate that the best Genetic Algorithm Models with Markov Order 1 in Figures 22, 9, 24, and 25 have better CV and better ICOMP scores than the "Restrict Ground Truth" subsets. This provides additional evidence that the best performing optimal subset is larger than four sensors. However, comparing the "Top 10 Sensor" results with the same "Restricted Ground Truth" results further reinforces that solving for a small subset directly is better than using our ranking procedure to select a small set of variables.

In summary, if one wishes to find the best performing optimal subset, it is generally computationally infeasible because computing sensor subsets with four sensors takes three hours, five sensors takes three to four days, six sensors takes 75 days, and seven or more sensors takes years. However, one can produce reasonably good approximations using our approach. On the other hand, if one is interested in solving for a small optimal subset and one has enough computing resources, then it is best to compute it directly.

6.7 Summary of Findings

The results presented in Sections 6.1, 6.2, 6.3, and 6.4 show that the Genetic Algorithm with the $ICOMP(IFIM)$ model criteria as the fitness function is able to find better models than the Stepwise Selection method. In addition, these sections show that the best models were found with Markov Order 3, and that setting missing values equal to zero is better than removing sensors that have missing values. Applying our voting technique to the Genetic Algorithm models allows us to find a better model (Figures 22, 24, and 25) than the best single Genetic Algorithm model (Figures 5, 13, and 17) on House 1, House 3, and across all houses. Therefore, on future homes we recommend comparing the best single Genetic Algorithm model with the model made from our voting process, and then selecting the best performing model from these two. However, if one is interested in finding the best model for a sufficiently small sensor subset, e.g., upto 5 sensors, it is recommended that one solve for this best model directly, because it should be computationally feasible to test all possible subsets.

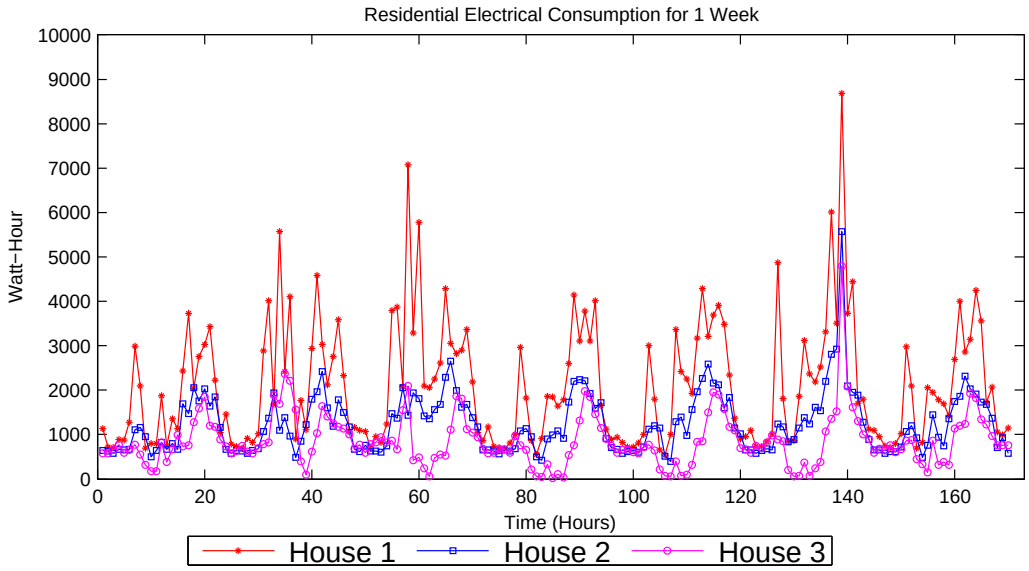


Figure 26: This figure presents one week of electrical consumption for all three residential homes, from the second week in September, 2010.

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 on the exact same schedule. Figure 26 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 [33], the SVR implementation is from LIBSVM [6], the HME implementation uses modified software provided by the authors of [25], and all remaining learning systems are implemented using existing MATLAB modules provided by Mathworks. Considering the reasonable performance

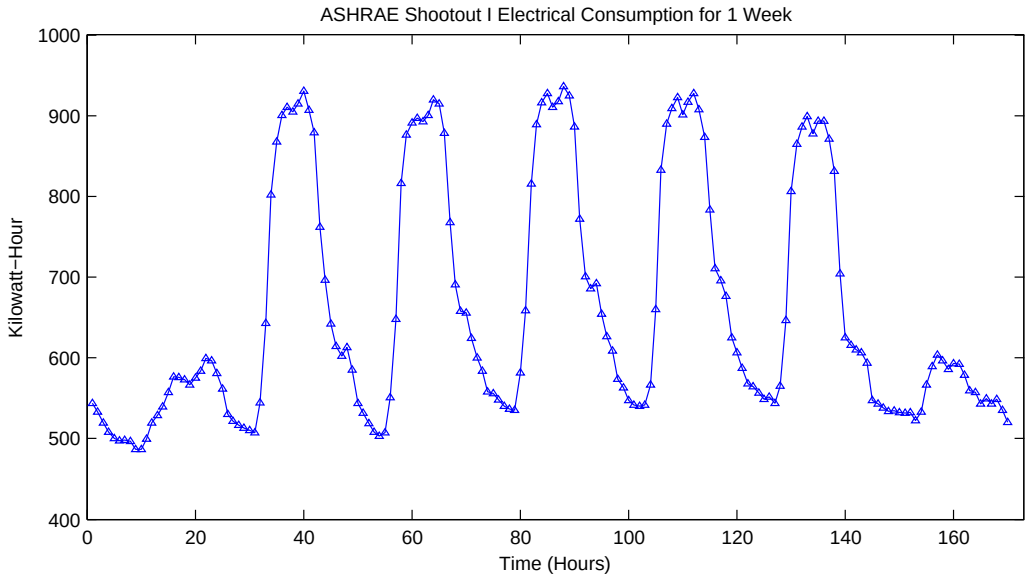


Figure 27: This figure presents one week of electrical consumption for the Great Energy Prediction Shootout building, from the second week in September, 1989.

by most techniques on the Great Energy Prediction Shootout data set and the fact that all techniques are built using established software, the only possible cause for not matching the established CV range is each house's complex energy usage patterns and the physical differences in the buildings.

Comparing the residential electrical consumption (Figure 26) with the commercial electrical consumption (Figure 27), shows that commercial buildings have fairly stable usage patterns and less sudden change. 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 28 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 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.

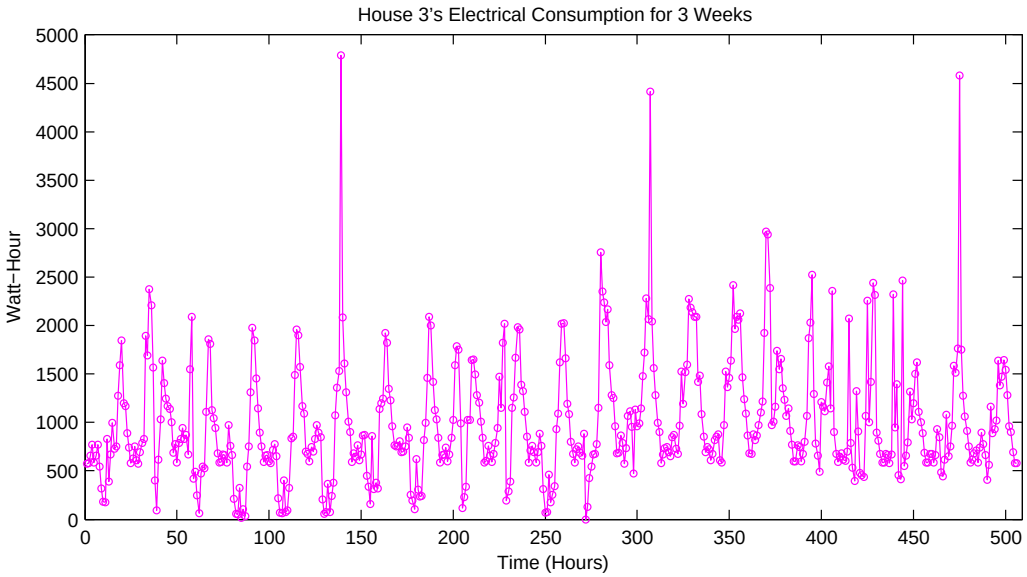


Figure 28: This figure presents three weeks of electrical consumption for House 3, starting from the second week in September, 2010.

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, which is more challenging than the standard 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 significant improvement on House 3: HME with LSSVM 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

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
LSSVM	8.85±0.18	0.02±0.21	6.95±0.21
RVM	8.87±0.16	-0.03±0.08	7.19±0.15
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**
HME-LSSVM	**8.91±0.23**	**0.02±0.20**	**7.00±0.19**
FCM-REG	10.50±0.27	0.02±0.27	9.00±0.25
FCM-FFNN	**8.74±0.26**	**0.05±0.24**	**6.99±0.21**
FCM-LSSVM	8.82±0.17	-0.00±0.13	6.97±0.23

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
LSSVM	3.77±0.44	-0.07±0.08	2.13±0.20
RVM	2.33±0.15	0.01±0.04	1.51±0.01
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**
HME-LSSVM	3.89±0.42	-0.08±0.09	2.21±0.18
FCM-REG	3.48±0.35	0.00±0.21	2.22±0.09
FCM-FFNN	**2.17±0.17**	**0.01±0.11**	**1.38±0.00**
FCM-LSSVM	3.88±0.46	-0.11±0.09	2.17±0.21

Table 13: 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.

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 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 LSSVM 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. Clearly the LSSVM approach learns a model that 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 LSSVM method builds a model that expects the testing response to resemble the observed training response. However, in this situation the electrical consumption pattern changes and the LSSVM 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 13). More importantly, LSSVM 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 LSSVM is the best technique for modeling each residential home. In addition, our results show that the previously accepted method, FFNNs, performs worse than the newer techniques explored in this work: HME-FFNN, LSSVM, and FCM-FFNN. Lastly, our results show that SVR and LSSVM perform almost equally with respect to CV and MAPE. However, experiments with SVR present poor MBE results, which makes LSSVM 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 LSSVM is the worst performing technique for the Shootout data set, and that shuffling the data improves its performance.

In addition, we aimed to determine which sensors are most important for predicting whole building energy consumption for the next hour. We demonstrated that a Genetic Algorithm used with the $ICOMP(IFIM)$ multi-objective criteria function is able to reduce model complexity, while still giving a reasonable goodness-of-fit. Additionally, we illustrated that the Stepwise Selection method is sometimes capable of producing smaller sensor subsets than the Genetic Algorithm approach, but the Stepwise Selection models are rarely less complex than the models generated by the Genetic Algorithm, even when the Genetic Algorithm includes additional sensors within the model. We introduced a method for ranking the sensors by combining all best models found from the Wrapper techniques, which was able to produce the best models for House 1, House 3, and across all houses. Additionally, using the ranking techniques and Wrapper methods we were able to illustrate some of the effects missing values had on the algorithms. Stepwise Selection performed better when all missing values were set to zero, and the Genetic Algorithm method was fairly indifferent to the missing data approaches. However, it found its best results generally when missing values were set to zero. Lastly, we compared our approach against the best possible subsets up to size four, which showed that it is computationally infeasible to directly compute a large enough subset that approximates the true best subset. Therefore, the Genetic Algorithm method is the ideal approach for future sensor subset selection.

In future work, we will explore sensor selection with the other machine learning techniques presented in this report. The selected sensors are dependent upon the machine learning technique that is being used to optimize the $ICOMP(IFIM)$ criteria function. Therefore, it is imperative to explore the solutions proposed by the other prediction techniques presented within this report. In addition, exploring sensor selection with the best performing technique, LSSVM, will hopefully provide the best sensor subsets. These subsets should help offset future cost for other building studies by reducing the number of installed sensors.

References

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Proceedings of the Second International Symposium on Information Theory*, pages 267–281. B.N. Petrov and F. Caski, eds., Akademiai Kiado, Budapest, Hungary, 1973.

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

[3] H. Bozdogan. Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, 1987.

[4] H. Bozdogan. Intelligent statistical data mining with information complexity and genetic algorithms. *Proceeding of JISS 2003, Lisbonne*, 2:15–56, 2003.

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

[6] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] Jeff Christian, A. Gehl, P. Boudreaux, J. New, and R. Dockery. Tennessee Valley Authoritys Campbell Creek Energy Efficient Homes Project: 2010 First Year Performance Report July 1, 2009August 31, 2010. pages 1–126, 2010.

[8] D. Crawley, J. Hand, M. Kummert, and B. Griffith. Contrasting the capabilities of building energy performance simulation programs. *Building and Environment*, 43(4):661–673, April 2008.

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

[10] R.H. Dodier and G.P. Henze. Statistical analysis of neural networks as applied to building energy prediction. *Journal of solar energy engineering*, 126:592, 2004.

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

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

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

[14] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[15] Pablo H. Ibargüengoytia, Luis E. Sucar, and Sunil Vadera. Real Time Intelligent Sensor Validation. *IEEE Transactions on Power Systems*, 16(4):770–775, November 2001.

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

[17] M.I. Jordan and R.A. Jacobs. Hierarchies of Adaptive Experts. *Advances in Neural Information Processing Systems 4*, pages 985–993, 1992.

[18] M.I. Jordan and R.A. Jacobs. Hierarchical Mixtures of Experts and the EM Algorithm. *Neural computation*, 6(2):181–214, 1994.

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

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

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

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

[23] C.A.M. Lima, A.L.V. Coelho, and F.J. Von Zuben. Pattern classification with mixtures of weighted least-squares support vector machine experts. *Neural computing & applications*, 18(7):843–860, 2009.

[24] D.J.C. MacKay et al. Bayesian nonlinear modeling for the prediction competition. *Ashrae Transactions*, 100(2):1053–1062, 1994.

[25] D.R. Martin, C.C. Fowlkes, and J. Malik. 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, 2004.

[26] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[27] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, 11(2):416–431, 1983.

[28] G.W. Rumantir. Comparison of second-order polynomial model selection methods: an experimental survey. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 980–980. JOHN WILEY & SONS LTD, 1999.

[29] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

[30] R.R. Schaller. Moore's law: past, present and future. *IEEE Spectrum*, 34(6):52–59, June 1997.

[31] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

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

[33] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least squares support vector machines*. World Scientific Pub Co Inc, 2002.

[34] J.A.K. Suykens, De Brabanter J., Lukas L., and Vandewalle J. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1-4):85–105, 2002.

[35] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

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

[37] V.N. Vapnik. An overview of statistical learning theory. *Neural Networks, IEEE Transactions on*, 10(5):988–999, 1999.

[38] A.S. WEIGEND and S. SHI. Hidden markov experts. In *Quantitative analysis in financial markets: collected papers of the New York University Mathematical Finance Seminar*, volume 2, page 35. World Scientific Pub Co Inc, 2001.

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