Predictive HVAC Control Modeling and Implementation of Deep Q Learning for Energy Usage Optimization

Daniel Izadi
Carnegie Mellon University
Pittsburgh, PA
dizadi@andrew.cmu.edu

Carolyn Goodman
Carnegie Mellon University
Pittsburgh, PA
cgoodman@andrew.cmu.edu

Abstract

This paper outlines the procedures for modeling the chilled water return temperature and chilled water flow rate of the Gates and Hillman Centers' HVAC system. Support Vector Machines (SVM) produced the most reliable test accuracy when implemented. SVM was also used to build a predictive model for the apparent power consumed by the building based on HVAC system data. These models were then implemented into a simulation of the HVAC control system. A deep reinforcement agent was trained to control the times at which the pumps were on or off with the goal of optimizing electricity consumption. The results showed that a neural network trained on the simulation saved 77.8 KWh of total energy and reduced peak power consumption by 7% compared to the actual output of the system for the day it was trained on. This is a promising result considering the agent was only allowed to control one aspect of the system's dynamics.

1. Introduction

Reducing energy consumption in buildings is an important step to reduce reliance on fossil fuels. Machine learning algorithms can be used to both predict building operations as well as optimize them. Because most renewable energy sources are intermittent, predicting energy usage can be used to implement energy storage as well as load shifting to reduce demand charges. Predictive models can also be used to help with maintenance to detect equipment issues (if operations are differing from the predictive model). In addition, having an electricity consumption predictive model can help create more informed Power Purchase Agreements for large consumers (e.g. Carnegie Mellon University) especially with the changing climate. Predictive models can be more accurate than standard building energy simulations as they are trained to actual building data rather than constrained to the building energy simulation software's setup. For this reason, a predictive machine learning model is a useful tool for training and testing reinforcement learning models prior to physical implementation.

Heating, Ventilation, and Air Conditioning (HVAC) systems are inherently reactive - different parts of the system will turn on when a threshold is reached. System operations can be better controlled using reinforcement learning models. Our project investigates the implementation of a reinforcement learning model to optimize the chilled water request of the Gates and Hillman Centers Chilled Water System. The chilled water request is currently set point controlled but can be better controlled via predicting environment changes and potentially pre-cool the building prior to peak loads.

Our predictive models were based on outdoor air conditions and humidity and airflow in HVAC zones. In addition, chilled water requests and the chilled water return temperature were used to predict the chilled water flow rate which was then used to predict the power consumption of the building. SVM was used for predictions due to its high effectiveness for non-linear data and we were able to get highly accurate models for the data tested.

The SVM models were used as the driving equations for a simulation of the HVAC control system. A neural agent was trained on this simulation with the ability to control a binary on/off decision within the system. The result was that the agent was able to develop a control strategy that consistently outperformed the actual system in both training and cross validation testing. This resulted in not only a reduction in electricity consumption but also peak power consumption for the Gates and Hillman Centers.

2. Related Works

The viability of our project rests on the accuracy of the predictive model, therefore, investigating how others have predicted energy consumption in buildings was our first step. Other research primarily focuses on predicting holistic energy consumption rather than the operation of individual components within the building. Due to air conditioning making up about 45% of the energy consumption in a commercial building, a lack of data on other energy consuming devices in the building may not be needed [1]. Ahmad et al. [1] reviewed support vector machine (SVM) model applications for building energy predictions and their advantages and disadvantages. Due to SVM's high accuracy with nonlinear data, it is a good option for fitting highly complex HVAC data. An SVM model to predict cooling load for an office building had a higher accuracy than the conventional back-propagation neural network. [2] presents a workflow for building energy consumption predictions, information on data preprocessing, and evaluation metrics. [3] used support vector regression to predict individual component power consumption, including the chiller, chilled water pumps, air handling unit fans, and air handling unit load. All models had over a 0.998 R^2 value. They also recommended the use of SVR models for building power consumption due to their speed and simplicity when compared to neural networks.

For the reinforcement learning side of our project, we investigated if buildings have reinforcement learning control algorithms in place for their HVAC system. [3] used genetic algorithm to search for the optimal outlet temperature of chilled water, air flow, and chilled water flow based on the predictive model results. Because this paper utilized a predictive model for training the genetic algorithm, it validates our decision to use predictive models to train our reinforcement learning model.

Different methods using reinforcement learning to optimize HVAC control have been outlined and discussed in [4]. The general dynamics of HVAC control using reinforcement learning is shown in Figure 1.

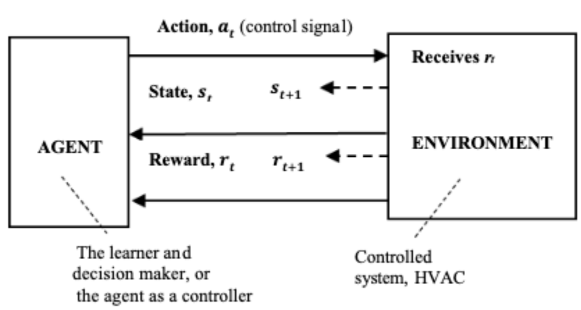

Figure 1. Basic reinforcement learning scenario for HVAC control [4].

Deep Q reinforcement learning models operate by optimizing a policy via a state-action function. With an optimized policy, the agent will choose the action at any given state that will maximize reward [4]. With an optimized policy for HVAC control using Q learning techniques the HVAC controller can act proactively rather than reactively. That is, the classical control policy of set point deviation for the activation of subsystems may benefit from an optimized policy with Q learning.

The key difference between previous models and our model, is that rather than predicting energy consumption exclusively, we are also predicting the intrinsic controls of the system while holding part of the system's operations constant. By isolating the control action to one subsystem, we not only reduce calculation time, but also validate that the control action recommendations via the neural net indeed produce positive results.

3. Data

Implementing our machine learning algorithm required a real and reliable data source. By using data from the Carnegie Mellon Facility Management Services, we were able to test on real data and have the ability to apply our models on more time varying data in the future.

3.1. Gates-Hillman HVAC Data Acquisition

PI Coresight is a web-based data visualization system that presents real-time and historical operational data collected from various data sources including SCADA and DCS equipment. Carnegie Mellon University's Facility Management Services offers complete access to the SCADA system's data points via PI Server and visualization and data analysis with PI Vision and the Excel plug-in PI DataLink [5]. The PI DataLink Excel plug-in allows the user to browse data from various buildings and campus wide systems (e.g. campus steam system and snowmelt system) and select data points to monitor. Historical data is selected by specifying the time range and time interval for the PI data points chosen. For our project, we focused on the Gates and Hillman Centers because it offered the most comprehensive data on the building's HVAC system.

Data acquisition required specifying all data points available for the HVAC system at Gates as well as the outdoor air conditions (temperature, humidity, etc.). This resulted in over 480 data points per observation. We also included the building's apparent power reading which is the optimization parameter for our neural network. A power reading is not available for the HVAC system exclusively but the total power consumption for the building is heavily correlated to HVAC operations. Our preliminary training and testing were based on 1-minute data collected from August 26 to August 29, 2019. A summer month with school in session was chosen as electricity consumption from the building's chilled water system would be at it's peak for the year.

3.2. Data Processing

Although the data acquired via the Excel PI DataLink plug-in was well organized, some data processing was necessary prior to training. Data point name lengths were reduced or renamed so they in a more human readable format. An "Entry" column was added to identify the timestep (minute of the day) so that the dataset would be ready to implement into the neural network later in the model and also because most of the variables are heavily time dependent. Because we only focused on a small section of year, date was not necessary for the model. The chilled water request (the action variable in the neural network) raw data point was not available via PI DataLink, we artificially created it by adding a binary feature that was 0 when the chilled water flow rate was $0 - gpm$ and 1 when the chilled water flow rate was greater than $0 - gpm$. This is in accordance with the control narrative for the Gates building chilled water system. We also removed observations with no data reported for features and removed features that remained constant throughout all observations as they would have no correlation to the selected label.

4. Methods

The methodology for our models was split into to different subsections. The first part of overall model focused on creating accurate predictive models of specific components of the chilled water system. The second part of the overall model focused on creating a deep reinforcement agent to control the chilled water system using the previously completed predictive models as the driving equations.

4.1. Predictive Modeling Methods

After processing the data, it was ready for implementation into the predictive model. The preliminary data set chosen consisted of 481 features and 5760 observations. Three different predictive models were necessary; chilled water return temperature, chilled water flow rate, and apparent power. Because our model is focused on modifying the chilled water request, the above three labels were chosen.

The chilled water return temperature indicates how well the chilled water system is performing for the current cooling demand. Chilled water return temperature is the temperature of the chilled water after it has cycled through the building; a higher chilled water return temperature indicates the need for a higher chilled water flow rate so it can provide more cooling to the building. The chilled water flow rate was predicted based off the chilled water request, the chilled water return temperature and the rest of the features in the model. The flow rate demonstrates how much chilled water is needed by the system and is highly correlated to the chilled water request. Therefore, the flow rate needed to be predicted at each time step in the reinforcement model discussed later in the paper. The last label is the apparent power which is the metric our algorithm is attempting to optimize.

Prior to training the data, we completed a train-test split with a 70:30 ratio. We then scaled the data to the training set. Because some of our features have very large magnitudes compared to others (e.g. air flow rate versus chilled water supply temperature), scaling the data was important to ensure accurate results. Scaling models were saved and reloaded to be applied to test data during experiments.

4.2. Feature Reduction and Feature Engineering

In order to reduce computation time as well as reduce the complexity of the model (and avoid inter-dependencies), the number of features was heavily reduced from 481 to 30. Because our model modifies the chilled water flow rate by changing chilled water request, we needed to ensure none of features used to predict chilled water flow rate were dependent on chilled water flow rate. Originally, we limited the features to outdoor air temperature and humidity along with the chilled water return temperature (which is also a label in a separate model) and chilled water request. Because we only modified the chilled water system, the air handling unit operation was set to remain constant. Therefore, humidity of each zone and air flow rate were added which chilled water return temperature and chilled water flow rate are heavily correlated to.

After performing feature selection, we trained the data and found that accuracy was low for the chilled water flow rate which is highly non-linear. The flow rate is dependent on how long the chilled water request has been on so we created a new feature for run time. This allowed the SVM model to more accurately predict the flow rate.

4.3. SVM Model

After feature engineering and establishing the predictive models necessary, we began training our data using Support Vector Regression. Based on the literature, SVR using the RBF kernal was deemed the best option for our application. We used scikit-learn's support vector regression toolkit [6] to train our model. In order to further validate use of SVR, we ran preliminary models using random forest and linear regression. Both models returned R^2 values for training

Table 1. SVR Final Model Details

	Chilled Water Return Temperature	**Chilled Water Flow Rate**	**Apparent Power**
SVR Kernal	RBF	RBF	RBF
C	1500	140	5000
γ	$\frac{1}{n_{features}}$	0.6	$\frac{1}{n_{features}}$
R^2	Train: 0.997 Test: 0.982	Train: 0.934 Test: 0.722	Train: 0.975 Test: 0.973
MSE_{test}	0.0014%	-	0.078%

data below 0.7. The final SVR models are outlined in Table 1 along with accuracy metrics.

Optimization of C, epsilon, and gamma was done for each model to achieve maximum accuracy. The chilled water flow rate test R^2 value is not ideal and appears to be overfitted. However, after performing multiple iterations with varying C, epsilon, and gamma values, this model is the best option for our data set. Future work would include adding more features to obtain a more accurate model for test data but we were limited on computation time for this project.

4.4. HVAC System Simulation Methods

The HVAC system simulation was made using the guidelines for OpenAI Gym custom environments [7]. These guidelines allowed for the simulation to be seamlessly passed to the neural network framework. A flow chart for the mechanics of the simulation working with the neural network are shown in Figure 2.

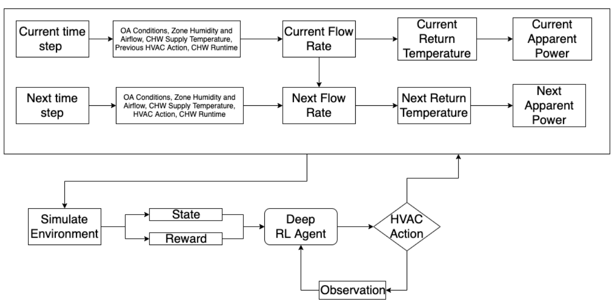

Figure 2. Flow chart of system simulation dynamics using SVM Model with Neural Network actions.

The system is passed a data frame with 30 state variables for each time step from raw data as well as columns for flow rate, feedback temperature, HVAC action, and power consumption which are initialized with the raw data for the first time step.

At its core, the system simulation uses the previous time step's state variables and feedback temperature to predict the chilled water flow rate at each time step. The flow rate and previous HVAC action, along with the other 30 state variables, are then used to predict the feedback temperature of the current time step. The power consumption at that time step is then predicted with the 33 state variables including HVAC action, flow rate and feedback temperature. These predictions are all made using the SVR models previously discussed.

At each time step, the neural agent controls the "HVAC Action" state variable. This essentially allows the agent to turn on or off the pumps in the Chilled Water System. The agent is then rewarded based on the resulting state of the system. The process is then repeated for the subsequent time steps until the end of the simulation. Each simulation represents a day and each time step within the simulation represents a minute in time.

Decreasing the change time between time steps would improve the accuracy of the SVR models and subsequently the simulation. This would also potentially improve the neural network's performance as it would be allowed to make more decisions that could potentially optimize the energy usage in a day.

4.5. Neural Network Design and Architecture

The neural network was designed for this application using the Keras Toolkit. The neural agent that was used was DQNAgent which uses deep Q learning algorithms to develop a policy that achieves maximum reward. Multiple neural network frameworks were tested with this agent by manipulating the number of layers, layer input and output sizes, and activation functions. The framework that provided convergence with the least amount of episodes is shown in Table 2.

Table 2. Neural Network Architecture.

Layer	**Type**	**Input Size**	**Output Size**	**Activation Function**	**# Parameters**
Input	Dense	33	64	Relu	33
Hidden 1	Dense	64	128	Relu	2,112
Hidden 2	Dense	128	256	Relu	8,320
Hidden 3	Dense	256	2	Relu	33,024
Output	Dense	2	1	Relu	514
Total Parameters	43,970				
Optimizer	Adam Function				
Compiler Loss Function	MSE				

There are 5 total layers in this neural network; an input layer, an output layer, and 3 hidden layers. The input layer size was 33 while the output layer size was 2. The first hidden layer had a layer size of 64, the second hidden layer had a layer size of 128, and the third hidden layer had a size of 256. The activation function used for all layers was the Relu function. The agent used Mean Squared Error (MSE) compiler loss function and the Adam function optimizer.

The neural network presented provided convergence in 500 episodes. It is important to note every framework tested converged to the same or similar final reward function value with more episodes.

4.6. Reward Function

The goal of this project was to minimize energy usage while maintaining proper control of the HVAC system. Deep Q learning algorithms rely heavily on the reward function to learn, thus the reward function needed to reflect our project goals to ensure success. The reward system was designed to reward when the feedback temperature was within the allowed deviation from the set point as well as reward energy savings. The agent is rewarded after each time step in the simulation.

Temperature Control Reward

It was necessary to ensure that the agent would properly maintain the feedback water temperature for the system to function properly. If the feedback temperature was within the bounds for deviation from the set point, the agent would receive a positive reward. If the feedback temperature was not within those bounds it would receive a negative reward.

It was also necessary to ensure that the agent did not deviate to feedback temperatures that would be harmful to the system or create an environment that was uncomfortable for occupants. The agent would receive a 100X negative reward if the feedback temperature was outside of a larger bound from the set point.

$$\text{R(t)} = \begin{cases} R(t-1) + r & \text{if } T_l < T_r < T_h \\ R(t-1) - r & \text{if } T_r \leqslant T_c \text{ or } T_r \geqslant T_h \\ R(t-1) - 100 * r & \text{if } T_r < T_C \text{ or } T_r < T_H \end{cases}$$

Where R_t is the reward at time step t, r is the baseline reward given, T_r is the return temperature at time step t, T_l and T_h are the low and high boundary set point deviations, respectively, T_L and T_H are the minimum and maximum return temperatures for the system to operate correctly, respectively.

Energy Consumption Reward

The agent must also be rewarded for saving electricity throughout the day. The agent would receive a negative reward if the power consumption at each time step was more than the raw data power consumption for that time step. Conversely, the agent would receive a positive reward if the power consumption was less than the raw data power.

$$\text{R(t)} = \begin{cases} R(t-1) + r & \text{if } P(t) < P_{baseline}(t) \\ R(t-1) - r & \text{if } P(t) > P_{baseline}(t) \\ 0 & \text{if } P(t) = P_{baseline}(t) \end{cases}$$

Where $P(t)$ is the power consumption and $P_{baseline}(t)$ is the baseline power consumption from the raw data at time step t.

5. Experiments

Verification and experimentation of the predictive models, system dynamics simulation, and neural network implementation was performed in three phases. First, multiple verification methods were implemented to ensure the accuracy of the system simulation and its driving equations, the SVR models. Once the simulation accuracy was verified a neural agent was trained using the simulation. Finally, the trained neural agent was implemented in a different day's HVAC data to cross-validate its strategy.

5.1. Simulation Validation Methods

The HVAC system simulation was validated using two methods. The first was to input the action of interest (i.e. Chilled Water Request) from raw data into the simulation. This allowed for comparison of the flow rate, feedback water temperature, and power consumption predictions between raw data and SVR models. Figure 3 shows the results for this validation method.

The results from this validation method allowed us to show the accuracy of our model and establish a baseline reward for the neural network to beat.

A simple control strategy was then implemented into the simulation as a sanity check. This strategy simply consisted

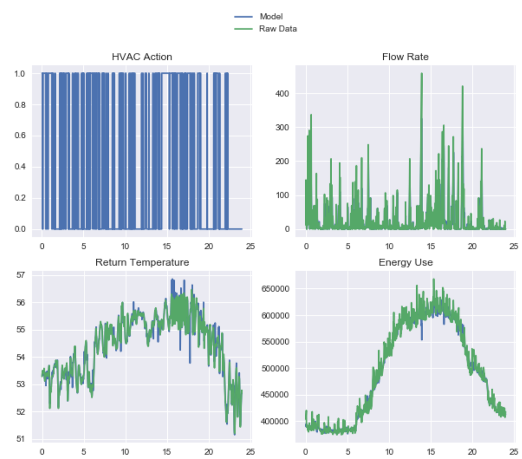

Figure 3. System simulation results with ground truth actions compared to raw data.

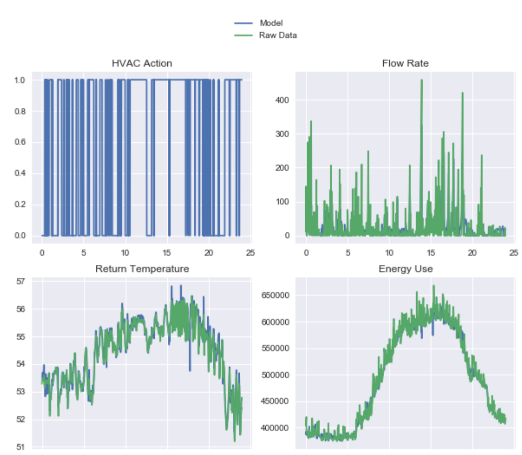

Figure 4. System simulation results simple control strategy implementation.

of the action was "ON" if the feedback water temperature was outside of a 1 degree threshold from the set point. Otherwise the action was "OFF". The expected outcome was that the simple control strategy would perform worse than the complicated control strategy from the actual system. The results are shown in Figure 4.

As shown, the simulation behaved as expected for both validation methods. There is very little discrepancy between raw data and simulation when the ground truth action was taken. This is further verified by the accuracy results for the SVR model described in Section 4.3. The model also behaved as expected for the simple control strategy. The strategy scored slightly worse than the baseline reward and the actions were similar to the ground truth actions.

5.2. Neural Network Implementation

The neural agent improved the reward function 15.3% from start to finish using Deep Q learning. The final reward value that was achieved after 500 episodes of training was -7,000. This can be compared to the -8,260 reward baseline that the actual actions achieved in the simulation. The agent saved 77.3 kWh of energy and decreased peak power consumption by 46.95 kW with the new strategy by changing this one control action. This equated to 0.65% of the total energy consumption for the entire building and a 7.02% decrease in peak power. A summary of performance comparisons between the Neural Agent's control strategy and the actual control strategy are presented in Table 3. Table 3 also includes the reduction in carbon dioxide emissions over that one day [8].

Table 3. Summary of performance metrics for neural network control strategy and actual control strategy.

	Energy Consumption	**Peak Power Consumption**
Actual Control Strategy	11.97 MWh	668.18 kW
Neural Agent Strategy	11.89 MWh	621.23 kW
Total Reduction	77.3 kWh	46.95 kW
Percent Reduction	0.65%	7.02%
Carbon Emissions Reduction	120.46 lbs	

It was noted that the control strategy developed by the neural agent minimized power consumption at times of peak power. Commercial entities typically get billed based on peak power consumption in the form of demand charges, thus this strategy would have monetary value if implemented.

Visual representations of the performance of the Neural Agent's control strategy after 500 episodes is presented in Figure 5. The agent's reward versus number of training episodes is presented in Figure 6.

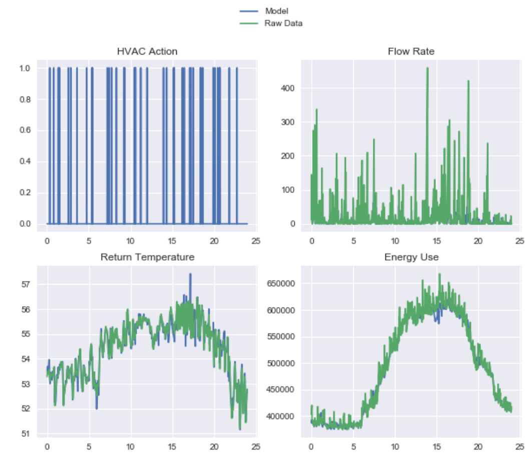

Figure 5. Neural network control strategy and resulting HVAC control parameters.

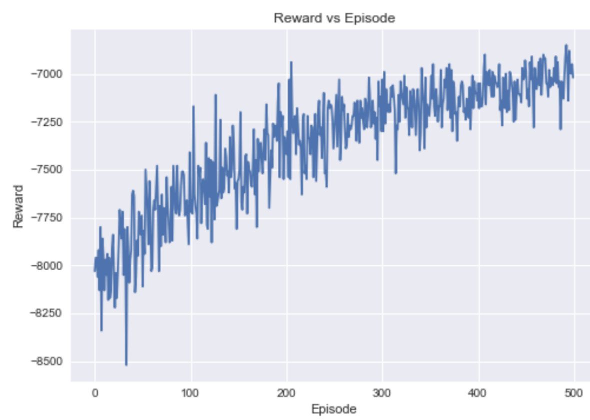

Figure 6. Reward as a function of number of episodes for the proposed neural network.

5.3. Neural Network Strategy Cross Validation

In order to test the performance of the neural agent's strategy, the agent was given the raw data frame for a different day. The agent was allowed to make control action decisions based on this new data. This was to verify that the agent's control strategy was not specific to the day it was trained but rather could be generalized to different environments. The agent's reward value in this test can be compared to the baseline reward for the raw data actions for that day. The results for the cross validation testing are shown in Figure 7.

The agents reward value was -9,890 and the raw data reward value was -11,920. The control strategy developed by

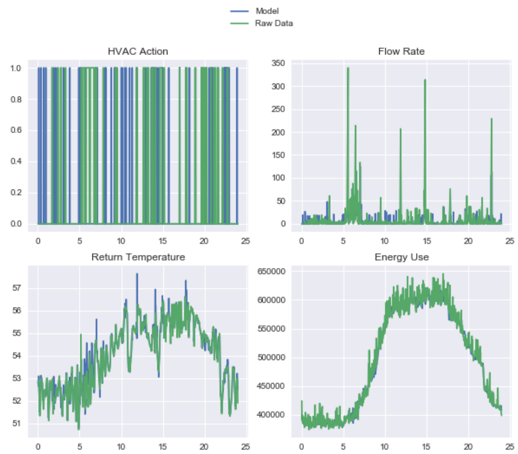

Figure 7. Cross validation testing results.

the neural agent was able to outperform the actual control strategy for the test date. It was also noted that the control strategy implemented by the neural network is very similar to the actual control strategy on that day. The energy reduction performance metrics for the cross validation test date as well as the daily emissions reduction are summarized in Table 4.

Table 4. Cross validation test performance results.

	Energy Consumption	**Peak Power Consumption**
Actual Control Strategy	12.01 MWh	645.37 kW
Neural Agent Strategy	11.95 MWh	617.78 kW
Total Reduction	65.3 kWh	27.61 kW
Percent Reduction	0.55%	4.46%
Carbon Emissions Reduction	101.82 lbs	

This result is promising considering the agent was only trained on one day's data. It would improve our model to train the agent on many days with different conditions throughout a year to develop an optimal and generalized control strategy.

6. Conclusions

An HVAC system simulation was built using SVR models for feedback water temperature (chilled water return temperature), chilled water flow rate, and power as the driving equations. This simulation produced accurate results when compared to the actual system dynamics. The strategy developed by the neural network improved the total energy usage in the simulation by 77.3 kWh and reduced peak power consumption by 45.95 kW when compared to the actual energy use on the day it was trained. The agent's strategy was cross validated by testing it on a different day in the same week as the training date. The cross validation results indicate that the agent's control strategy is applicable across different environments.

The agent did not change how the pumps operated, but rather when they operated to achieve this result. This was done by allowing the neural agent to control a binary "ON" or "OFF" decision in the Chilled Water System. The result is promising in that there was only one control action modified by the neural agent to produce this result.

Methods for building the SVR models, HVAC system simulation, and neural network architecture are outlined in this paper. Future works may include training the predictive model and reinforcement model across many different days and months for a more generalized strategy. In addition, the reinforcement model can be trained for more episodes to ensure convergence (although we believe our models converged). We can also train our models with a finer data set (every second instead of every minute). In order to ensure more realistic operations of the recommended control algorithm, future works could also incorporate more constraints for the chilled water request operation and eventually implement the algorithm into a real system to truly test outcomes.

References

[1] A review on applications of ANN and SVM for building electrical energy consumption forecasting | Elsevier Enhanced Reader.

[2] Jens Schneider, Matthias Dziubany, Anke Schmeink, Guido Dartmann, Klaus-Uwe Gollmer, and Stefan Naumann. Chapter 8 - Predicting energy consumption using machine learning. In Guido Dartmann, Houbing Song, and Anke Schmeink, editors, *Big Data Analytics for Cyber-Physical Systems*, pages 167–186. Elsevier, January 2019.

[3] Ching-Wei Chen and Yung-Chung Chang. Support Vector Regression and Genetic Algorithm for HVAC Optimal Operation, 2016.

[4] Ivars Namatēvs. Deep reinforcement learning on hvac control. *Information Technology and Management Science*, 21:29–36, 12 2018.

[5] Carnegie Mellon University. hvac-intro-page - Facilities Management Services - Carnegie Mellon University.

[6] sklearn.svm.SVR — scikit-learn 0.21.3 documentation.

[7] Christopher Hesse. How to create new environments for gym, April 2019.

[8] OAR US EPA. Greenhouse Gases Equivalencies Calculator - Calculations and References, August 2015.