Interreg
2 Seas Mers Zeeën
INCASE
European Regional Development Fund

Machine learning perspectives for smart buildings: an overview

John Klein*

*Department of Electrical Engineering and Automation,
Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL - Centre de Recherche, en Informatique, Signal et Automatique de Lille, F-59000 Lille, France

May 17, 2017

Abstract

Smart home or smart building design relies on the combination of several hard and software new technologies. Concerning the software part, machine learning algorithms play a pivotal role as they allow the system to make autonomous decisions.

In this report, we review several machine learning techniques that have been tested and implemented in smart buildings. We discuss their benefits and limitations through several use cases that are in line with the objectives of the INCASE project.

1 Introduction

Smart buildings is an applicative field which aims at designing intelligent systems that can control and manage buildings with as little human supervision as possible. Any type of building is eligible for such technologies: homes, offices, factories, airports, schools and so on. Each building type may have its own specificities but the core idea is the same. In short, the building needs to be equipped with sensors and actuators. Each such device must be connected to a central server through an appropriate network technology and protocol. Finally, a software is run on the server. The software task is to collect all data coming from the sensors, analyze them and produce decisions as to when and how actuators should be triggered.

Concerning the software aspects of smart buildings, artificial intelligence algorithms are key components of the decision system. Among the vast artificial intelligence literature, machine learning has proved to solve complex decision making tasks with remarkable accuracy. Machine learning is a research field that attempts to reproduce learning principles from animal psychology into programming code. In general, the machine is given a dataset containing input-decision pairs. The algorithm exploits these data to predict decisions when new inputs are given.

In this report, we review state-of-the-art machine learning techniques that have been proposed and tested in the context of smart buildings. We start with a quick reminder on machine learning paradigms and recall several notions in this field. Since smart building systems usually require to solve several tasks, we review several use cases separately. The rest of the document is therefore organized with respect to each use case.

Smart factories and technical building management are two fundamental questions addressed as part the INCASE project. Although some of these use cases are not always directly related to these questions the devices implemented may provide information allowing to address other tasks. For instance, INCASE is interested in power consumption monitoring but power consumption can be useful to detect abnormal behavior within the building. In addition, all intelligent systems are built using the same machine learning building blocks, therefore all the solutions discussed in this report can be transposed to INCASE related problems. For instance, we discuss human detection using imaging sensors. This technology can be used track the activity of staff members inside a factory and provide assistance to them. It can also be argued that visual understanding algorithms are key elements for this application. Such algorithms can be used as well to detect motions that could be used to trigger lights or production sub-systems.

2 Machine learning basics

Generally speaking, machine learning aims at producing algorithms that have a predictive power. One can reach that goal by exploiting data and optimizing an objective function that features the impact of prediction errors as well as prediction successes. As said in the introduction, the mathematical and algorithmic mechanisms that are employed in machine learning are inspired from learning principles in animal psychology. This means that the decisions of the systems are compared to the correct ones and that the algorithm updates itself so as to avoid reproducing incorrect decisions in the future.

The learned models allow to solve multiple tasks that can give the feeling that the machine has its own reasoning capabilities. Actually, this is of course fictional. All the decisions produced by the machine are direct consequences of humanly programmed computation. In other words, machines do not make decisions on their own they just obey code lines.

In this section, we review machine learning paradigms and recall important related notions. For more comprehensive presentations of machine learning, reference books such as [4, 25] may be browsed.

2.1 Supervised learning

A supervised learning problem starts with a dataset. Each data in this set is a pair

$$\left(\mathbf{x}^{(i)}, y^{(i)}\right),$$

for i from 1 to n. In this pair, $\mathbf{x}^{(i)}$ is called an example and $y^{(i)}$ is called an output (or decision). The algorithm must learn the functional relation f_0 between any example $\mathbf{x}$ and its outputs $y = f_0(\mathbf{x})$.

Example 1. Suppose a customer applies for a credit card in a bank. The bank has some personal information about the customer (salary, account balance, total debt, age, gender and so on). This set of information pieces is an example $\mathbf{x}$. The corresponding output is binary (allow or deny the card)

Each piece of information inside an example is called a feature. The space where examples live is usually referred to as feature space. Most of the time, this space is a vector space and we can use linear algebra calculus. Sometimes examples contain categorical data like the customer gender in our example. However, categorical data can be embedded into a vector space. Consequently, we will make the assumption that any example $\mathbf{x}$ is a vector in $\mathbb{R}^p$ in the remainder of this document.

The integer p is the dimensionality of the examples while n is the number of examples. In modern machine learning, a famous bottleneck is termed big data. This means that n is very large and typically, the dataset cannot be loaded on a single computer memory. When p is large, one speaks of tall data.

Supervised learning is subdivided into two categories: classification and regression. Classification deals with categorical outputs like the accept or deny decision. In this case, y live in a set $\mathcal{C}$ that is deprived of any mathematical structure. Regression deals with continuous outputs living in a vector space. For instance, if the bank wishes to determine the maximal mortgage that can be granted to the customer, then this becomes a regression task.

Since our goal is to determine function f_0, the solution space has an infinite dimensionality. Most of the time, people resort to parametric models, that is, candidate functions are in bijective correspondence with a parameter vector $\boldsymbol{\theta} \in \Theta$ and the space Θ has finite dimensionality. Even if this simplifies the problem, this latter is still ill posed in the sense that they are infinitely many functions $f_{\boldsymbol{\theta}}$ such that

$$f_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right) = y^{(i)}, \forall i.$$

This established fact implies that learning algorithms must resort to additional constraints to converge to a solution. Usually, the chosen parametric model will not allow the estimated function $\hat{f}$ to have chaotic variations. The machine learner has done a good job when a trade-off has been found between predictive function complexity and data fit. An excessive data fit is a situation called overfitting while an excessively simple decision function is a situation called underfitting.

2.2 Unsupervised learning

In the unsupervised learning paradigm, the learner is also given a dataset, but the dataset contains examples only. The outputs are missing. In the worst case, the number of classes into which examples must be sorted is not known either.

The only available information is proximity between examples. Indeed, if two examples are close in the feature space, the usual assumption is that they belong to the same class. When data points seem to aggregate around an attraction pole, they supposedly are member of the same class. This problem is also known as clustering.

Of course, unsupervised learning is doomed to perform poorly as compared to supervised learning. Unfortunately, acquiring labeled data (examples with outputs) is costly and difficult. Imagine someone wants to design a medical image processing software that can detect cancer signs. Only physicians have the appropriate skills to assign labels to such examples. Consequently, one needs to employ expensive staff for a boring repetitive task.

In many applicative context, only a subset of the dataset is labeled. This framework is known as semi-supervised learning. Another possibility is to give the learner a budget of $n' < n$ examples for which he, or she, is given the correct output. In this setting, the learner queries the environment and therefore this paradigm is called active learning.

2.3 Reinforcement learning

The last paradigm that we evoke in this report is reinforcement learning [31]. This paradigm is dramatically different as compared to the previously mentioned ones.

In this setting, the learner is not given a dataset, but it will be possible for him to build one of its own to some extent. Instead the learner is given the possibility to try any input and obtain a reward in response to that input. The goal here is also to learn a function, but this function is meant to lead a commanded system to a desired state. The learned function is often referred to as the policy function. It should drive the system so that the desired state is achieved in the smallest number of moves possible.

The motives behind reinforcement learning is connected to control theory. This paradigm is especially interesting in the smart building context for buildings contain several controllable systems that we wish to regulate automatically (heating, air conditioned, lights, door access, etc.). Reinforcement learning is thus an interesting perspective as part of the INCASE project.

3 Anomaly detection

One of the most important goals of the INCASE project is to equip factories or technical buildings with many sensors and monitoring technologies. Suppose for simplicity that each sensor delivers a real 1D signal and that one records each signal for a given time span t_f. Suppose also we have m sensors. We can concatenate all these times series in a $m \times t_f$ matrix $\mathbf{X}^{(i)}$. If we collect several instances of such matrices, we can build an unlabeled dataset.

Although this is not the prime goal of these sensors, a general profile can be learned from this dataset. When we observe a new instance $\mathbf{X}$ that is significantly different from previously seen ones, then we can deduce that something potentially dangerous is happening. For instance, if power consumption is twice bigger than usual for several minutes then maybe a machine is malfunctioning and this could cause a mechanical accident or a fire outbreak.

In [30], the author processes daily electrical power consumption only. He extracts features from each time series (average consumption, peak demand, etc.). The feature vectors are then sorted with respect to day types (Monday, Tuesday, and so on). Afterwards, outlier data points are sought inside each subset feature by feature. A statistic is computed to compare the absolute difference between the maximal feature value and its mean value. The outlier identification is performed by comparing the statistic to a critical value designed for this test [29] when data points are normally distributed. This test is known as generalized extreme studentized deviate (GESD).

A similar approach is introduced by Li *et al.* [21]. The extracted features are mean daily-energy consumption, peak daily consumption and two parameters that are part of an autoregressive model fitted to the time series. Outliers are identified among feature vectors using GESD when the sample size is large enough. When it is not, another method based on ordered statistics is used. This method is known as Q-test and relies data point discrepancies analysis (after they have been re-ordered in ascending order). The authors then uses Fisher linear discriminant analysis to classify each feature vector into a day type (week day or working/non-working day). Outlier examples are removed from the training dataset.

In [7], Chicco surveys different clustering algorithms as well as feature extraction methods from power consumption signals. However, the goal of the approach is to find groups of customers instead day groups.

Another possibility to detect anomalies is to mine for abrupt changes inside the signals. To the best of our knowledge, this lead does seem to have been paid much attention in the smart building literature. There are however many approaches proposed in the machine learning and signal processing literature that could serve this purpose.

A mixture of experts [16] model could be fit to each signal. This statistical model can fit signals undergoing different regimes. A latent random variable represents to which regime each signal sample belongs to. Each regime has its own model and parameters. All these parameters (including the mixture ones) can be learned using the expectation-maximization (EM) algorithm [9].

In a recent paper [6], Chamrouki introduces an approach that can model abrupt changes and cluster signals. Again, a mixture model is used and each mixture component represents a cluster. Within each cluster, a piecewise polynomial regression model is used to fit signals. The author also uses adapted versions of EM to learn jointly all the model parameters.

4 Activity recognition

The devices that can be installed in a factory or a technical building, as proposed in INCASE, can also be used for human activity recognition. Many kinds of activities can be tracked. As part of smart factories, specific tasks of workers can be identified to provide them with appropriate assistance. For instance, if the system is able to detect that the worker is trying to assemble two pieces together, then a pre-calculated torque can be applied to help him (or her) to tighten a fastening screw. A more common task is also human presence detection. Such information is important for security (production may be stopped if someone is inside a given area) and for quality engineering (presence of people may corrupt the quality of the production in clean rooms).

In this section, we provide a quick overview of the human detection use case. First, let us stress that most approaches are non statistical and learning free. They resort on deterministic electrical engineering solutions. For instance, a simple solution is to equip building users with an RFID chip [18]. If RFID antennas are deployed in the building, user positions may be easily tracked. The drawback of this approach is that building meshing is costly and users have to accept carrying chips.

Most of the time, occupancy detection requires anyway the installation of specific sensors. Whether the final objective is energy saving (as in INCASE) or not, low consumption sensors are preferred. In this vein, passive infrared sensors are an interesting solution. Human bodies emit photons in the infrared domain while non-organic matter do generally not. In particular, hot objects (like heaters) emit infrared photons therefore sensors should be placed accordingly to avoid false positives.

Agarwal *et al.* [2] introduce a human presence detection approach relying on passive infrared sensor signals. They also use switches to detect if an office door is open or not. When open, human presence is assumed. When closed, the decision is made from passive infrared sensor measurements. To avoid triggering a positive detection too soon, the measurements are analyzed after the first pulse plus six seconds. Pulse patterns from the sixth second to the eighth second are then used to decide on the office occupancy. From an algorithmic point of view, this approach is simplistic and many machine learning algorithms could be used to improve detection accuracy provided that one can access raw input data instead of binary data. Indeed, binary pulses are obtained after comparing the raw signal to a predefined threshold. The value of this threshold could be learned automatically as part a supervised technique such as support vector machines, logistic regression or linear discriminant analysis.

Other low cost presence sensors examples are pressure sensors to detect chair occupancy, temperature, humidity, CO_2 and acoustic sensors [27]. A slightly more costly solution consists in deploying a network of low resolution cameras. The cameras need to be low resolution and produce images at low frequency for both cost and power consumption reasons. The images also need to be processed and classified in an embedded computing system. In [13], the authors uses such a camera network. Images are processed so that background is suppressed and occupants thus produce blobs in the images. For indoor cameras, background suppression is not a costly operation. Blobs are groups of connected pixels with a significantly high average value. The authors also centralize occupancy data and try to learn dependencies between occupancies of neighbor rooms using a multivariate Gaussian model.

Note that temporal dependencies are also interesting to assess as done in [23] where a hidden Markov model is used to that end. Dodier *et al.* [10] also take temporal dependencies into account as part of a Bayesian network. They use several redundant passive infrared sensors. The state transition probabilities are learned in a frequentist fashion. The architecture of the network is not learned but is derived from an analysis of causes and effects relations between the random variables.

More information on this topic can be found in the survey by Nguyen and Aiello [28].

Outside the smart building literature, human detection has also raised a lot of interest. In particular, human detection in images and videos is a vivid research topic. A recent survey [26] covers the corresponding state-of-the-art. The major problem with the approaches presented in this community is that they rely on high resolution cameras whose specifications do not meet our energy saving requirements.

In the computer vision community, deep learning algorithms have proved to obtain remarkable level of accuracy in scene understanding tasks. In this family of machine learning algorithms, popular models for video processing are recurrent neural networks (RNN) and long-short term machines networks (LSTM) [5, 11]. These models are adapted for processing time sequences. Roughly speaking, they are neural networks implemented in a recursive fashion meaning that the output of the network at time t is the image of a linear combination of recent output history through a non-linearity (which is usually a sigmoid function). The weights involved in the linear combination encode the influence of past decisions in the present one. The network could be unrolled across time to yield a vanilla neural network but the parameters of this unrolled network would be tied.

Although these algorithms are very appealing in their performance promises they usually demand high resolution images, large training datasets and considerable computation resources. Nonetheless, there are reasons to believe that deep nets is an interesting perspective for smart building too for two reasons:

- most of the computation effort is paid at training time not at test time. Training can be performed on standard machine and the trained network may be uploaded to the embedded system attached to the low cost camera.
- the occupancy detection task is much simpler than visual understanding tasks studied in computer vision. Smaller resolution images and smaller datasets may be enough for this task.

5 Speech recognition for command design

In this section, we provide a few useful references dealing with natural language processing and, more precisely, on speech recognition. Speech recognition has many potential applications for smart factories and other INCASE related topics. For instance, the production could be supervised using voice commands which is much more ergonomic for a human operator than clicking on graphical interfaces or typing commands in a console.

This is the only topic discussed in this report for which the industrial context does not imply particular specifications as compared to other contexts. In factories, there may be an increased need for robustness as this is a very noisy environment. Again, the algorithms should not consume too much energy and computation resources. But these two are not really limitative and the core ideas of natural language processing do apply.

Speech recognition is a multi-level pattern recognition challenge. The input data is a 1D signal. Depending on the semantic complexity of the information encoded in this signal, the goal of the machine may be to identify words or sentences. Each word is itself composed of a set of phonemes which are basic sounds that human beings can produce. Each phoneme is a small time series containing several signal samples. Depending on the sampling frequency and on the speaker, phonemes (and consequently words) have variable lengths.

The signal processing community has produced several contributions allowing to compute feature vectors from raw signals. These feature vectors are an intermediate representation of the data. If each training example $\mathbf{x}^{(i)}$ is turned into a feature vector $\mathbf{z}^{(i)}$, then usual supervised learning techniques can be employed on the dataset $\left\{\left(\mathbf{z}^{(i)}, c^{(i)}\right)\right\}_{i=1}^{n}$. The signal processing contribution relies on the invariance of the feature vectors to speaker variability and background noise.

Before the deep learning breakthrough, the most popular feature extraction approach for speech signals was Mel Frequency Cepstral Coefficients (MFCCs) [8]. This method starts with a short term Fourier analysis, supposing that phonemes have approximately constant time span. Each local Fourier representation is given as input to a filter bank to evaluate the energy carried along different frequency intervals. Human cochlea (an organ in the ear) cannot disambiguate nearby frequencies. The filter bank mimics the human hearing system. This disambiguation difficulty is more prominent in high frequencies therefore the frequency interval lengths are not constant. Afterwards, one must take the log of filter output energies. This is also motivated by the human hearing system for which loudness is not in linear scale. Finally, we take the discrete cosine transform of the log energies to decorrelate the feature vector entries.

Observe that an MFCC feature vector represents only one phoneme. Since words are phoneme sequences, one needs a time dependency model. The standard choice is hidden Markov models. In a more recent approach, Graves *et al.* use MFCC feature vectors on deep recurrent neural network. This network combines the convolutional spirit of deep nets with the recurrent aspect of LSTMs.

Nowadays, some deep learning algorithms have outperformed MFCC based approaches on a number of challenging datasets. Hinton *et al.* [15] use a deep belief net (DBN) on feature vectors that are filter bank outputs. This net is obtained by stacking generative probabilistic models known as restricted Boltzmann machines (RBMs). This net can learn a feature representation on its own. Its parameters are first obtained in an unsupervised training phase called pre-training. The output of the DBN is then connected to a deep neural net which is trained in a supervised fashion as usual.

In summary, the main perspective of speech recognition for the problems addressed as part the INCASE project is just to adapt existing tools in the machine learning literature and not to develop a new one.

6 Automatic building regulation

Activity detection discussed in section 4 is often a building block of automatic building regulation which is a higher level task. There are many things that can be regulated in a building but most of the reported work in this field focus on heating, ventilation and air conditioned (HVAC).

Regulation consists in computing relevant inputs (or command signals) to a system so that the output of the system reaches a predefined level known as the reference signal $\mathbf{r}$. This task is the main motivation that lead to the creation of control theory. Usually the system output is not the only one that one wishes to control. All the variables for which control is needed are stacked in a vector $\mathbf{s}$ called the state vector.

Unsurprisingly, most of the contributions that can be found in the smart building literature addressing regulation resort to control theoretic approaches and a physical model of the system. Afram and Janabi-Sharifi [1] published a nice review in which they discuss different solutions for controller design for HVAC: linear (PID) , non-linear, model predictive, optimal or fuzzy logic based controllers. An example of such controllers is given in [12]. The author recalls the differential equations behind the HVAC system physics. In this example they involve temperature, pump energy and water flow rate. The controller is obtained by optimizing an objective function. It thus belongs the optimal controller family.

There are long to date connections between control theory and machine learning. One possibility is to train a neural network to obtain a regressor that acts as a controller. Examples of such neural network controllers for HVAC are given in [17, 32, 19].

Another strong connection between machine learning and control theory is reinforcement learning. In this setting, the learner is given a set of actions that he (or she) can choose from. Each action modifies the state $\mathbf{s}$ and, in this context, the reward is proportional to the inverse of $\|\mathbf{r} - \mathbf{s}\|$. The set actions basically consists in increasing or decreasing the command signal by some constant value. The goal is to find the set of actions that will achieve $\|\mathbf{r} - \mathbf{s}\| = 0$ as fast as possible and stabilizes around this state. The set of actions is given by a policy function π which is what we need to learn.

Examples of reinforcement learning solutions for HVAC control are given in [14, 22]. In the broader context of smart buildings, other relevant references are [20, 3]. The standard model for these approaches are Markov decision processes (MDPs) or extensions of it. This model formalizes the notions of actions, policy function and state. It also specifies a number of deterministic or probabilistic relations between those. The most popular learning algorithm is Q-learning which is able to converge to the optimal policy.

The application of reinforcement learning in smart building environments remains quite rare. Many developments have been introduced in the machine learning community over the past decade. In particular, deep reinforcement learning [24] algorithms are a promising perspective that is likely to perform well in intelligent building control.

7 Conclusions et perspectives

In this report, we have highlighted that machine learning techniques have potential benefits in almost every aspects of smart factories and buildings. The INCASE project aims at studying solutions relying on artificial intelligence that can provide relevant assistance to operators and managers in production sites or technical buildings.

We have focused on the following use cases: anomaly detection in data-streams coming from connected devices, building occupancy detection, triggering system by vocal commands and building control. In each of these use cases, we provide innovative perspectives and references involving machine learning. We hope that they will serve as discussion starting points for next steps in INCASE WP2.

References

[1] Abdul Afram and Farrokh Janabi-Sharifi. Theory and applications of {HVAC} control systems – a review of model predictive control (mpc). *Building and Environment*, 72:343 – 355, 2014.

[2] Yuvraj Agarwal, Bharathan Balaji, Rajesh Gupta, Jacob Lyles, Michael Wei, and Thomas Weng. Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 1–6. ACM, 2010.

[3] Heider Berlink and Anna HR Costa. Batch reinforcement learning for smart home energy management. In *IJCAI*, pages 2561–2567, 2015.

[4] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] Wonmin Byeon, Thomas M Breuel, Federico Raue, and Marcus Liwicki. Scene labeling with lstm recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3547–3555, 2015.

[6] Faicel Chamroukhi. Piecewise regression mixture for simultaneous functional data clustering and optimal segmentation. *Journal of Classification*, 33(3):374–411, 2016.

[7] Gianfranco Chicco. Overview and performance assessment of the clustering methods for electrical load pattern grouping. *Energy*, 42(1):68 – 80, 2012. 8th World Energy System Conference, {WESC} 2010.

[8] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[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*, 39:1–38, 1977.

[10] Robert H. Dodier, Gregor P. Henze, Dale K. Tiller, and Xin Guo. Building occupancy detection through sensor belief networks. *Energy and Buildings*, 38(9):1033 – 1043, 2006.

[11] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[12] B. Dong. Non-linear optimal controller design for building hvac systems. In *2010 IEEE International Conference on Control Applications*, pages 210–215, Sept 2010.

[13] Varick L Erickson, Yiqing Lin, Ankur Kamthe, Rohini Brahme, Amit Surana, Alberto E Cerpa, Michael D Sohn, and Satish Narayanan. Energy efficient

building environment control strategies using real-time occupancy measurements. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 19–24. ACM, 2009.

[14] Gregor P. Henze and Jobst Schoenmann. Evaluation of reinforcement learning control for thermal energy storage systems. *HVAC&R Research*, 9(3):259–275, 2003.

[15] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[16] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[17] Soteris A. Kalogirou. *Artificial Neural Networks and Genetic Algorithms for the Modeling, Simulation, and Performance Prediction of Solar Energy Systems*, pages 225–245. Springer London, London, 2013.

[18] Younghun Kim, Zainul Charbiwala, Akhilesh Singhania, Thomas Schmid, and Mani B Srivastava. Spotlight: Personal natural resource consumption profiler. In *Proceedings of the 5th Workshop on Hot Topics in Embedded Networked Sensors*. Citeseer, 2008.

[19] Andrew Kusiak and Guanglin Xu. Modeling and optimization of hvac systems using a dynamic neural network. *Energy*, 42(1):241–250, 2012.

[20] D. Li and S. K. Jayaweera. Reinforcement learning aided smart-home decision-making in an interactive smart grid. In *2014 IEEE Green Energy and Systems Conference (IGESC)*, pages 1–6, Nov 2014.

[21] X. Li, C. P. Bowers, and T. Schnier. Classification of energy consumption in buildings with outlier detection. *IEEE Transactions on Industrial Electronics*, 57(11):3639–3644, Nov 2010.

[22] Simeng Liu and Gregor P Henze. Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis. *Energy and buildings*, 38(2):148–161, 2006.

[23] Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. The smart thermostat: Using

occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 211–224, New York, NY, USA, 2010. ACM.

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[25] K. P. Murphy. *Machine Learning, A Probabilistic Perspective.* MIT press, 2012.

[26] Duc Thanh Nguyen, Wanqing Li, and Philip O. Ogunbona. Human detection from images and videos: A survey. *Pattern Recognition*, 51:148 – 175, 2016.

[27] Tuan Anh Nguyen and Marco Aiello. Beyond indoor presence monitoring with simple sensors. *PECCS*, 2012:5–14, 2012.

[28] Tuan Anh Nguyen and Marco Aiello. Energy intelligent buildings based on user activity: A survey. *Energy and Buildings*, 56:244 – 257, 2013.

[29] Bernard Rosner. Percentage points for a generalized esd many-outlier procedure. *Technometrics*, 25(2):165–172, 1983.

[30] John E Seem. Using intelligent data analysis to detect abnormal energy consumption in buildings. *Energy and buildings*, 39(1):52–58, 2007.

[31] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[32] M Zaheer-Uddin and N Tudoroiu. Neuro-pid tracking control of a discharge air temperature system. *Energy Conversion and Management*, 45(15):2405–2415, 2004.