NANYANG TECHNOLOGICAL UNIVERSITY

MACHINE LEARNING BASED DATA ANALYTICS FOR IOT DEVICES

XU HANG
SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING
2017

MACHINE LEARNING BASED DATA ANALYTICS FOR IOT DEVICES

XU HANG

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Master of Engineering

2017

Acknowledgement

First and foremost, I would like to express my sincerest gratitude to my supervisor Prof. Yu Hao for his countless support and invaluable help in my research and study during these years in Nanyang Technological University. I'm really fortunate to work with his supervision. His profound knowledge, technique insights, patient guidance and constructive comments have been quite helpful for me to address research issues. He taught me how to explore a research issue deeply and how to focus on work under high pressure. Both in research and life, he set a great example for me: work hard all the time, have passion for both work and life, and have courage to face challenges. I sincerely thank him for the research opportunity and really appreciate the research experience with his supervision.

Secondly, I would like to appreciate my confirmation exercise examiners Prof Er Meng Joo, Prof Lim Meng Hiot, Prof Arindam Basu for assessing my project. I would like to thank EEE Graduate Program Office and VIRTUS IC Design Centre of Excellence lab staffs Mrs. Lim-Tan Gek Eng, Mr. David Robert, Miss Christina Wong and many others who have been very friendly and patient to help me on administrative matters. I would like to express gratitude to Energy Research Institute @ NTU for supporting my research during master study period. I would like to thank Dr. Koh Leong Hai, Mr. Krishnamoorthy Baskaran, Mr Venkat Arumugam for their kind help and support during project development.

Thirdly, I would like to express my sincere thanks to my wonderful colleagues, Huang Hantao, Cai Yuehua, Suleman Khalid Rai, Huang Xiwei, Ni Leibin, Liu Zichuan, Jiang Yu, Liu Xu, Yang Huan, Liu Wenye and many others who accompanied me to walk through my most memorable years. I would like to express special thanks to Huang Hantao who taught me and help me a lot in research issues. I also would like to thank Cai Yuehua especially, who has developed the indoor positioning system and were willing to helped me with the technical details.

Last but not least, I owe my deepest gratitude to my dearest Dad, Xu Zhaoxin, mom, Han Jiaojun for their endless love and unconditional support. I would like to give special thank to my boyfriend who had faith in me and accompanied me all the time. Thanks for being with me on each step of my life. They are the greatest motivation for me to a higher level.

Abstract

The Internet of Things (IoT) is a new emerging Eco-system of networked devices with the potential of delivering significant benefits to individuals and societies. By incorporating of IoT system into human lives, physical world information can be actively collected and analyzed to facilitate the decision making process towards ultimate benefits. To perform data analytics and smart decisions, machine learning based algorithm becomes popular for IoT system. In this thesis, machine learning based algorithm has been applied to IoT application in smart home/building field and biomedical field. From machine learning algorithm perspective, the thesis first has studied single hidden layer neural network (extreme learning machine) for data regression. Then such neural network is further extended to multi-layer neural network (convolutional neural network) for more complex data analytics. From IoT application perspective, smart building energy management system and lensless blood-cell counting system have been designed and built with machine learning based data analytics.

- For modern residential buildings, the energy can be provided by main electricity power-grid as well as additional power-grid of solar energy. As it is common for main electricity power-grid to experience extremely huge load demand during the peak period, renewable energy, such as solar energy, should be optimally allocated for alleviating the effect of peak load demand on the main power-grid. Therefore, the development of smart building energy management system (BEMS) becomes strongly needed with ambient intelligence (AmI). Here, the AmI of BEMS refers to data analytics within an indoor environment in refsponse to occupants' presence. For current indoor AmI, the main challenge is how to response dynamic ambient changes in real time because of the latency of processing backend in cloud. Another challenge is how to deal robustness and scalability in large space by using a distributed solution. Therefore, a real-time data analytics platform with distributed architecture is needed for smart

building with AmI. Moreover, occupant behavior with random nature, which has significant effect on energy consumption in residential buildings, is also a challenge for load forecasting. It is necessary to proposed an efficient and feasible approach to detect and analyze the occupant behavior in real time.

To tackle aforementioned challenges, this thesis has proposed an on-line sequential machine learning based data analytics platform with distributed architecture. It can build and update forecasting model continuously without increasing computation complex due to incremental least-square solver. Then, occupant behavior is monitored by real-time indoor positioning system through Wi-Fi signal detection and analytics. Based on extracted occupant behavior profile and energy consumption profile, solar energy allocation problem can be solved which aims at decreasing the peak load demand in main electricity power-grid. The flow of the machine learning based data analytics for BEMS can be summarized: sensing real-time data, predicting occupant behavior profile and energy consumption profile, and allocating solar energy. Instead of using cloud, all these operations are running on the smart-gateway network with limited computational resource.

In the experiment, the proposed approach can achieve better in load forecasting accuracy by 14.83% comparing with support vector regression (SVR). Based on the extracted energy consumption profile and occupant behavior profile, solar energy can be allocated, achieving reduction of peak load by 15.20% and saving of energy cost by 51.94% with comparison of static prediction based allocation strategy.

- With the rapid development of biomedical IoT system, lensless blood-cell counting system (LBCS) has become a potential solution for blood cell counting in point-of-care testing (POCT) that can offer critical information for rapid on-site disease diagnosis and monitoring. Composed by the complementary metal oxide semiconductor (CMOS) and microfluidic channel, the LBCS has competitive advantage over conventional lens based imaging system due to mass production of inexpensive CMOS and rapid development of lab-on-a-chip technology. However, such a system can only capture low-resolution images with loss of detailed information in cell morphology because there is no optical lens. It is imperative to improve the image resolution through super-resolution (SR) processing in system-level for cell counting and recognition, For current SR technology, the main challenge is how to improve captured image resolution with low cost of processing resource and also without degrading throughput.

In this part of thesis, convolutional neural network based super-resolution (CNNSR), which is lightweight, feed-forward and potential in implementation of on-chip hardware, has been employed to tackle challenges mentioned above. Moreover, a lensless blood cell counting prototype is demonstrated, which not only utilizes custom designed backside-illuminated CMOS image sensors but commertial CMOS image sensors. Based on such a prototype, when the captured low-resolution cell image is input, a corresponding high-resolution image is reconstructed through CNNSR. In experiment, the image resolution can be improved by 4×. Comparing with another machine learning technique, extreme learning machine (ELM), CNNSR can perform better in resolution improvement quality by 9.5%. Such experiment results can demonstrate that CNNSR approach possesses the potential of high-efficiency and high-quality resolution improvement in the LBCS towards POCT.

In summary, the main contribution of this thesis can be summarized as follows. Firstly, to manage residential building energy consumption and optimize renewable energy source allocation yet without sacrificing occupants' comfort, an on-line sequential machine learning based data analytics platform with distribution structure has been proposed. It can build model with continuously update capability for both occupant behavior profile and energy consumption profile extraction. Considering the result of short-term load forecasting, solar energy allocation decision is made to reduce peak load and save energy cost. Using such a platform, data can be analyzed and predicted in real time with low computation complex. Secondly, to solve the low-resolution limitation of lensless microfluidic imaging for point-of-care testing blood cell counting, CNNSR is employed to recover high-resolution cell images from corresponding captured low-resolution images. Using such an approach, the cost of processing resources is low and the system throughput is not degraded, which is matched well with the developed lensless blood cell counting system.

Contents

List of Abbreviations

AI	Artificial Intelligence
AmI	Ambient Intelligence
ANN	Artificial Neural Network
BEMS	Building Energy Management System
BSI	Back-side Illuminated
CIS	CMOS Image Sensor
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
CNNSR	Convolutional Neural Network based Super-resolution
CPGA	Ceramic Pin Grid Array
DOS-ELM	Distribution of Online Sequential Extreme Learning Machine
ELM	Extreme Learning Machine
ELMSR	Extreme Learning Machine based Super-resolution
FPGA	Field-Programmable Gate Array
FSI	Front-side Illuminated
HAS	Home Automation System
HEMS	Home Energy Management System
HF	High-frequency

HR	High-resolution
HVAC	Heating Ventilation and Airconditioning
IoT	Internet of Things
IP	Internet Protocol
IPS	Indoor Positioning System
LBCS	Lensless Blood-cell Counting System
LR	Low-resolution
MAC	Management and Control Address
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
PD	Photodiodes
PDMS	Poly-dimethysiloxane
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RSSI	Received Signal Strength Indicator
SLFN	Single Layer Feed-forward Network
SR	Super-Resolution
SVM	Support Vector Machine

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The Internet of Things (IoT) is a new emerging Eco-system of networked devices with the potential of delivering significant benefits to individuals and societies. By incorporating of IoT system into human lives, physical world information can be actively collected and analyzed to facilitate the decision making process towards ultimate benefits. For example, an energy-efficiency IoT based smart home can be built by integrating energy sensors, environmental sensors and positioning sensors. Such a smart home IoT system can analyze energy price, indoor environment and occupant behavior in order to perform energy management yet with consideration of occupants comfort level. Another example is biomedical IoT system such as blood cell counting system which can perform disease diagnosis and medical treatment by utilizing advanced algorithm and big data analytics. In this thesis, two IoT systems, namely building energy management system (BEMS) and lensless blood-cell counting system (LBCS), are designed and investigated. To perform accurate data analytics and smart decision making, machine learning based algorithms have been applied to these two IoT systems.

1.1.1 Building Energy Management System

Energy plays a vital role in residential daily life, social-economic development and national security. Demand of electricity energy is exponentially increasing and the traditional energy sources, such as coal and oil, are depleting at an alarming speed. To alleviate the worldwide energy crisis, it is crucial to manage the usage of energy sources and improve the efficiency of energy consumption. As a large part of energy consumption in a power system, residential energy consumption can be optimized by Home Energy Management System (HEMS) [2].

HEMS is an IoT system based on devices interconnected by network communication. HEMS is mainly developed for monitoring, planning and controlling the energy consumption according to predicted energy demand at user side. Therefore, to accurately predict energy demand is vitally important for effective energy management. Numerous approaches have been developed to predict more accurate energy demand for HEMS, which can be mainly classified into artificial intelligence (AI) based models and statistical technique based models. Both of them are widely adopted for short-term load forecasting [3–10]. Recently, machine learning has been proven as a powerful approach for short-term load forecasting in [11, 12]. However, most models cannot be updated according to the dynamical changes of load pattern, which will increase the error probability of prediction. Therefore, the first challenge for residential building energy management is how to sequentially update model with new coming training data for responding dynamic changes in real time. Moreover, short-term load is fluctuate and random due to various influence factors such as temperature, humidity and occupants behavior, which will make it difficult in establishing function relationship between historical load sequence and future energy demand. Among numerous influenced factors, the biggest challenge of prediction accuracy is occupants behavior with random nature that is difficult to detect and monitoring. Furthermore, for modern residential buildings, the energy can be provided by main electricity power-grid as well as additional power-grid of renewable solar energy. As it is common for main electricity power-grid to experience extremely huge load during the peak period, renewable energy should be optimally allocated to alleviate the effect of peak load on main power-grid. Therefore, the development of BEMS becomes strongly needed with the challenge how to deal robustness and scalability in large space by a distribution solution.

To tackle aforementioned challenges, this thesis has proposed an on-line sequential machine learning based data analytics platform with distributed architecture. It can build and update forecasting model continuously without increasing computation complex due to incremental least-square solver. Then, occupants behavior is monitored by real-time indoor positioning system through Wi-Fi signal detection and analytics. Based on extracted occupant behavior profile and energy consumption profile, solar energy allocation problem can be solved which aims at decreasing the peak load demand in main electricity power-grid. The flow of the machine learning based data analytics for BEMS can be summarized: sensing real-time data, predicting occupant behavior profile and energy consumption profile, and allocating solar energy. Instead of using cloud, all these operations are running on the smart-gateway network with limited computational resource.

1.1.2 Lensless Blood Cell Counting System

For rapid on-site disease monitoring and diagnosis, blood cell counting in point-of-care testing (POCT) is vitally important, which can offer critical information [13,14]. Existing techniques for blood cell counting can be summarized as two types: one is manual counting through high magnification optical microscopy with high numerical-aperture objective lenses, the other is automated counting via commercial flow cytometers. However, for manual counting, accuracy is greatly effected by operator experience and the process is time-consuming with low throughput. And for automated counting, commercial flow cytometers are really expensive due to bulky and sophisticated optics. Both of these two techniques are thereby prohibitive for POCT. Recently, rapid development of microfluidics-based lap-on-a-chip technology make it possible for the miniaturized microflow cytometer of portable flow cytometry [15, 16]. The inexpensive CMOS image sensors (CIS) can be integrated underneath the microfluidic channel to develop a microfluidic-based lensless imaging system [17, 18]. With a white light source illuminating from above, the contact image of cell or the direct projected shadow can be captured by the underneath image sensor without lenses. However, as there is no optical lens, these captured cell images are low-resolution without details about cell morphology information. Therefore, some system-level solutions should be proposed to improve the resolution of cell images, such as super-resolution (SR) [19].

SR aims at reconstructing high-resolution (HR) images based on corresponding low-resolution (LR) images [20]. Existing SR approaches mainly include multi-frame SR processing and single-frame SR processing, which are both applicable for lensless imaging system. Since multi-frame SR processing requires large storage for continuously capturing, storing and processing LR images, it is not suitable for on-chip hardware implementation [17]. As the alternative method, single-frame SR approaches are thereby imperative [21]. However, some problems of previous single-frame SR methods can not be ignored, such as overly smooth for bilinear interpolation [22], intensive computation for some machine learning based SR methods [23–25], learning speed and optimization performance problems in [26].

In this thesis, aforementioned SR problems are tackled by employing an efficient machine learning based a single-frame SR approach that is called convolutional neural network based SR (CNNSR) [27]. Due to the lightweight and feed-forward characteristic, it is potential for on-chip hardware implementation. Moreover, an IoT system for lensless blood cell imaging and counting is built with a custom designed back-side illuminated (BSI) CIS as well as a commercial CIS. This proposed microfluidic lensless cell counting system can achieve high-quality super-resolution without limitation of throughput, which can offer portable and cost-effective solution for POCT. The whole processing flows can be described briefly as follows. First, HR image library, composed by HR cell images of different types, is off-line prepared to train a reference model. Then, with cell solution sample inputing into microfluidic channel, LR cell images are on-line captured and processed by CNNSR using reference model to reconstruct corresponding HR images. Finally, these recovered HR images can be output for medical counting and recognization.

1.2 Thesis Contributions

The contribution of this thesis can be summarized as below:

- This thesis proposed an online sequential machine learning based data analytics on distributed computational platform. Based on captured real-time occupant behavior information and energy consumption data, it can build a short-term load forecasting model with sequential update capability. Then, the result of short-term load forecasting is utilized to solve the solar energy allocation problem which aims at decreasing the peak load value in the main electricity power-grid.

- This thesis proposed an efficient machine learning based single-frame SR approach based on convolutional neural network for microfluidic lensless cell counting system. Such system can achieve high-quality super-resolution without limitation of throughput, which can offer portable and cost-effective solution for point-of-care testing system.

1.3 Organization

This thesis is organized as follows. Chapter 2 presents overview and literature review about IoT system and machine learning basics. Chapter 3 introduces distributed online sequential machine learning based data analytics for smart building energy management system. Chapter 4 describes the convolutional neural network based super-resolution for lensless blood cell counting system. Conclusion and future work are summarized in Chapter 5.

Chapter 2

Fundamental and Literature Review

2.1 Internet of Things

2.1.1 Overview

The term Internet of Things (IoT) was firstly proposed by Keyin Ashton in 1991 in an article about supply chain management [28]. With the evolution of technology in decades, the definition of IoT has been developed on the base of main goal that a computer can sense and analyze information without human intervention. The evolution of IoT system has been summarize by Li et al. [1] as shown in Figure 2.1. Gubbi et al. [29] summarized the definition of IoT without restriction of communication protocol: The interconnection of physical devices, computing devices and network connectivity can enable all the objects to share collected information over a unified network and perform common operation with data analytics in cloud computing.

The architecture design of IoT should take scalability, extensibility and interoperability into consideration [1]. As a mainstream IoT architecture, the service-oriented architecture (SoA) has been adopted widely [30–33]. The SoA can be described as follows [1]:

- Sensing Layer: Various sensors are deployed to automatically capture the status information of things and share data among IoT devices.
- Network Layer: The advancement of communication technology, mainly including wired and wireless communication medium, facilitates the rapid development

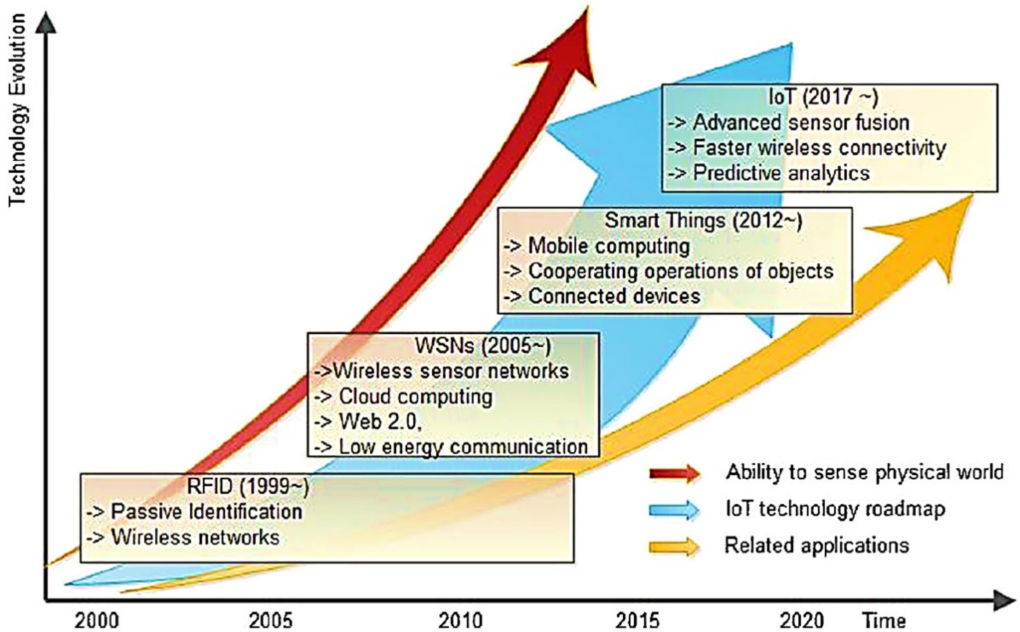

Figure 2.1: The evolution of Internet of Things [1]

of IoT. Based on network layer, all the IoT devices can be connected together, resulting in data aggregation from sensors and data transmitting to decision-making center for high-level services.

- Service Layer: Composed by application programming interfaces (APIs), the common requirements of applications and protocols, service layer can manage and provide service to users.

- Interface Layer: The interface layer can provide an effective interface mechanism to simplify the interconnection between IoT applications and users.

With rapid and great development of IoT, various areas have been benefited from the IoT application, such as industrial applications, smart building application, biomedical application [34], infrastructure, security and surveillance. Here, the smart building application and biomedical application will be introduced in details.

2.1.2 IoT for Building

(a) Overview

As an individual and home scale of IoT application, distribution of sensors and actuators in house is helpful to controlling home equipment for a better home energy management. For instance, the household heating equipments can be adjusted according to the weather and users' preference; the lighting appliances can be turned on/off based on the light intensity; household appliances can be switched off automatically when not needed. To manage residential energy consumption without sacrificing occupants' comfort, home energy management system (HEMS) has been become a popular research topic. HEMS can perform energy optimization and building automation by controlling an interrelated network of physical components. This system consists of home gateways, smart meters, sensors, household appliances and communication system, shown in Figure 2.2. Here, smart meter is a sensor device that measures energy consumption for minimizing the power theft. Home gateway is the core controller for household appliance scheduling and energy management.

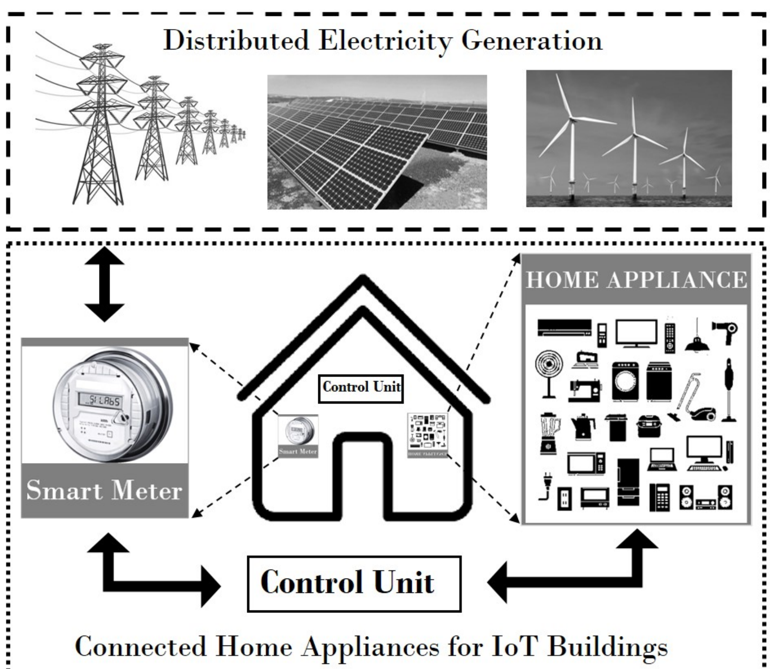

Figure 2.2: IoT based building integrated with smart grid

(b) Load Forecasting

Load forecasting plays a vital role in electric industry, which is helpful to make decisions on generating electricity, purchasing electricity power, allocating renewable energy source, switching load or developing infrastructure [35]. Generally, it can be categorized as short-term, medium and long-term load forecasting. The short-term load forecasting is utilized for hourly, daily or weekly energy forecasting, while medium-term are usually from a week to a year and long-term is even longer than one year. The accuracy of load forecasting is greatly decided by forecasting methods, mainly including statistical techniques or artificial intelligence (AI). For statistical techniques based forecasting methods, functional relationship can be established between load demand and effect variables. Generally, the generated forecasting function is linear, which has no capability to provide sufficiently accurate prediction results when there is non-linear relationship between effect variables and load data [36]. In contrast, the artificial intelligent-based forecasting methods, such as artificial neural network (ANN) [6–10], genetic algorithms [37] and fuzzy logic [38, 39], can model complex non-linear relationship between load data and influence factors.

For residential building energy management, short-term load forecasting is dominant in predicting daily load profile and peak load period, which is beneficial to decision-making of renewable energy source allocation. Therefore, short-term load forecasting approaches will be introduced in detail as follows: [35]:

- **Regression methods** are one of typical statistical techniques, which models the relationship between load data and influence factors. Papalexopoulos et al. [40] proposed a linear regression-based model for the calculation of short-term load forecasting, which completed innovative model building and robust parameter estimation. Haida et al. [41] presented a regression based daily peak load forecasting method with transformation technique. Charytoniuk et al. [42] performed short-term load forecasting through nonparametric regression in the form of a probability density function.

- **Time series methods** assume that there is internal structure in data, such as trend, seasonal variation or autocorrelation. It aims to exploit and analyze these internal structures. Classical time-series methods include autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA) and autoregressive integrated moving average with exogenous variables (ARIMAX).

Here, only taking time and load as input parameters, ARMA is generally utilized for stationary process while ARIMA is used for non-stationary processes. ARIMAX models also take weather, time of the day and other influence factors for load forecasting [43, 44].

- **Fuzzy logic** can associate an input with a certain qualitative ranges and then deduce output from fuzzy input variables, which is also one technique for mapping inputs to outputs. The advantage of fuzzy logic is that precise input and noise free input are not needed, and also a mathematical mapping model is not necessary. With such generic conditioning rules, the fuzzy logic system can perform load forecasting robustly [35, 45, 46]. Moreover, when precise output value is expected, a "defuzzification" process can be performed after logically processing fuzzy inputs.

- **Neural networks** have capability to perform non-linear curve fitting, which means that the output of ANN has a linear or non-linear relationship with the input. Feinberg et al. [35] demonstrated that back propagation is the most popular ANN architecture for load forecasting. Based on supervised learning and continuously valued functions, the back-propagation neural networks can determine input weights by matching historical data to desired outputs in the training section. For example, Bakirtzis et al. [47] proposed a fully connected three-layer feed-forward ANN for short-term load forecasting and utilized back propagation algorithm for model training. This model takes historical hourly load data, the day of week and temperature as input variables. Moreover, the ANN can be combined with other techniques for load forecasting, such as time series [48], fuzzy logic [49] and regression trees [49].

- **Support Vector Regression** (SVR) is a powerful technique in solving regression problems, which is typical one of machine learning methods. It can effectively solve non-linear problems with a small number of training data. Dong et al. [50] firstly utilized SVR to predict monthly power consumption and demonstrated that SVRs have a good performance in load forecasting. Li et al. [51] applied SVR to predict hourly cooling load for office buildings. Results showed that SVR performs better than the conventional back-propagation neural networks. Hou et al. [52] demonstrated that SVR performs better than ARIMA model in load forecasting for the heating, ventilation and air-conditioning system. These works shows that SVR has a good performance in predicting hourly, daily and monthly energy consumption profile.

In summary, each approach has its own advantages and drawbacks in certain cases. The statistical technique based methods are relatively easy to learn the linear mapping between input and output, but the model is not flexible enough and the outputs are lack of accuracy. Both ANN and SVR can perform well in non-linear problems so that they are more applicable to solve load forecasting problem with complex variables. Since ANN has some problems such as excessive training, tuning burden and unsatisfactory generalization performance, the SVR has better performance than ANN in many cases [53]. However, most SVR experiments were performed using a small quantity of training data [50–52]. For large amount of training data, the model training process of SVR will become extremely slow and complex.

To solve the computation complex and running time problem, extreme learning machine (ELM) is becoming popular for solving load forecasting problems [12, 54, 55]. Experimental results in these works demonstrated that ELM can perform well in load forecasting with relative lower computation complex and faster training process. Therefore, this thesis proposed distributed online sequential extreme learning machine (ELM) for daily load forecasting. Through the proposed approach, the model is generated by combination of several parallel training results and can be continuously updated with new coming training data.

(c) Effect of Occupant Behavior

Among various influence factors of load forecasting, occupant behavior can be regarded as one most significant reason of forecasting uncertainty due to its inherent complexity and random nature. The occupant behavior influences building energy consumption in direct or indirect ways, for example turning on/off heating, ventilation, and air-conditioning (HVAC) system, opening/closing windows, turning on/off lights [56]. Some studies have demonstrated that occupants' interaction with building components has a significant effect on energy consumption prediction. For example, [57] studied the effect of occupant behavior on building heating section and concluded different ways that occupants utilize the heating system influence greatly on heating energy consumption with obviously different consumption level. And [58] found that the difference of energy consumption can reach up to 150% when occupant behavior was taken into consideration in maximized and minimized way. Therefore, it is significant to take occupant behavior into consideration for load forecasting in building energy management system.

As shown in the Figure 2.3 [59], occupant behavior affects building energy management system through movement and actions, then further determines the environment factors and energy consumption. In turn, both environmental factors and energy consumption influence occupant behavior via economic, physiological and psychological factors. Since an individual occupant behavior is affected by psychological, physical and various factors, it is complex to thoroughly study occupant behavior in building level.

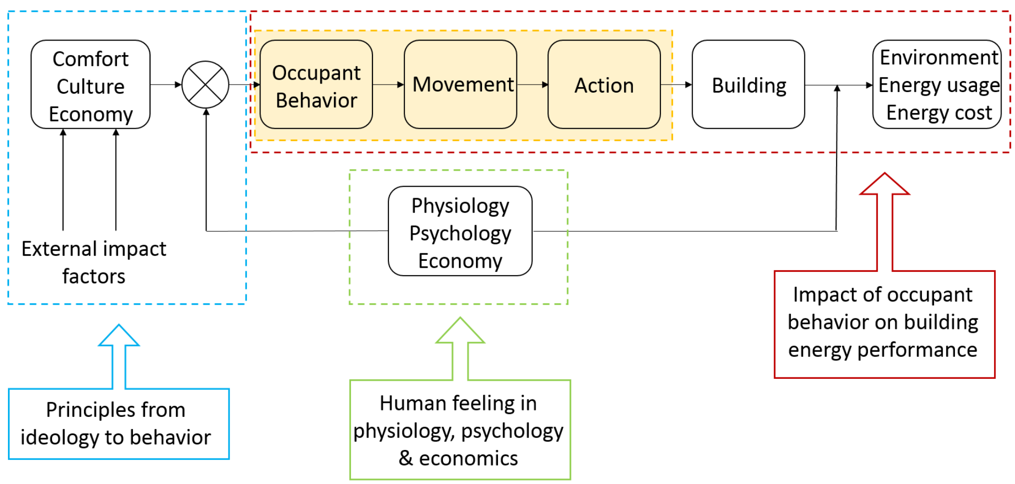

Figure 2.3: Scope of occupant behavior research in building energy management system

To solve this problem, many researchers narrow the modeling of occupant behavior down to a simplified representation with focus on occupants action and movement. Here, these actions, mainly including dimming/switching lights, opening/closing windows, adjusting HVAC or pulling up/down blinds, are influenced by multiple environmental factors as well as psychological and sociological factors. As for the effect of environmental factors, for example, occupants are usually driven to open/close the windows by multiple factors, including indoor air quality, indoor humidity, outside noise and outside weather conditions [60]. Pulling down blinds has a relationship with the magnitude of vertical irradiance while there is no clear relationship between pulling up blinds and the amount of solar radiation [61]. Also, opening/closing blinds is also influenced by psychological factors, for instance, occupants are likely to open blinds if they want to keep view and connection to outdoors [62] while the blinds are possible to be closed when occupants desire to obtain more personal private space [63].

Despite occupant action, occupants' movement is an another popular research topic in these years because occupants presence and movement are demonstrated as a preeminent factor affecting occupants behavior [59]. [63–65] indicate that the arrival, departure and duration of presence of occupants have a significant effect on building energy consumption. In this thesis, we detect and analyze occupant behavior in terms of occupant movement via Wi-Fi signal, which will be introduced in detail in Section 3.2.2.

2.1.3 IoT for Biomedicine

(a) Overview

Composed by medical sensors, wireless communication techniques and data processing center, the biomedical and healthcare IoT system can monitor medical parameters such as blood pressure and body temperature, and then send the real-time information to data computation center for medical analytics. If the captured information is analyzed out of normal range, rapid response action will be performed at once [1]. Recently, with the development of wearable sensors, patients can be monitored continuously in living environment and the captured information, such as burned calories and walking steps, can help patients to improve their lifestyle for preventing health problems [66]. Moreover, many works have been done for exploiting sensors and wireless computation techniques to improve bio-medical and healthcare IoT system [34, 67]. For example, mobile devices and personal computers can be used as wearable sensors and analytics center seperately, which leads to home-level healthcare. Therefore, such a system can provide high-quality and high-efficiency healthcare service.

(b) Image Super-resolution

As partial function of biomedical and healthcare IoT system, blood cell counting and recognization can help users to monitor their health status from a more professional perspective. To perform better diagnosis via blood cells, the resolution improvement of cell image is necessary. Image super-resolution (SR) is aimed at recovering one or more high-resolution (HR) images from one or more low-resolution (LR) images input, which has been a hot research field during last two decades. It has been employed in many applications, such as medical image processing [68, 69], face image improvement and recognition [70], satellite and aerial imaging [71]. However, the image super-resolution is an inherent ill-posed problem. Since the LR data is assumed to be obtained by low-pass filtering or down-sampling the corresponding HR data, the loss of high-frequency (HF) information will cause that different HR images can be degenerated into same LR images [72]. To solve the SR problem with under-determined nature, various methods and algorithms have been proposed and developed.

SR algorithms can be categorized based on the number of LR images involved: multi-image based SR algorithms and single-image based SR algorithms. Here, single-image SR based algorithms will be introduced in detail. For the single-image based SR algorithm, some algorithms will be employed to recover the missing information though the relationship between HR and LR images from a training database. Generally speaking, it can be categorized into two types: domain-specific single-image SR algorithm and generic single-image SR algorithm. The domain-specific single-image SR algorithm is focused on specific image types, such as graphics artwork, scenes and faces, while the generic single-image SR algorithm can process all kinds of images by utilizing typical properties of the original image as priors. Based on these image priors, the single-image based SR algorithm can be classified into four types of approaches [73]:

- Prediction Models: For prediction models, HR images are generated from LR images through a predefined mathematical formula without any training data. For example, the interpolation-based methods, such as bilinearing, Lanczos and bicubic, can generate HR pixel intensities through averaging neighboring LR pixel values. Such kind of methods can perform well in smooth regions because interpolated intensities have similar value as neighboring pixels. However, at high-frequency region or along edges, these models can not generate sufficient large gradients.

- Edge Based Methods: As an important original image structure, edge features are learned as priors, such as the parameter of a gradient profile [74] and the width and depth of an edge [75]. Such methods can generate high-quality edges in recovered HR images. However, as the priors are learned form edges primarily, other HF structure regions, such as textures, cannot be reconstructed effectively.

- Example Based Methods: For the example based methods, examples can be obtained from external datasets [76], internal datasets [24] and combined sources [25]. The external datasets are formed by a great number of natural images, resulting in that external-example based methods can be applied to any type of input images while the generated HR image appearance is plausible without specific features. Then the internal dataset is composed by patches extracted from example images with common properties. Since these patches are quite relevant to the type of input LR images, those specific features can be recovered in the output HR images. However, this internal-example based method cannot be applied to other kind of input due to the limited and small amounts of patch pairs.

Moreover, based on a set of paired HR and LR training examples, the mapping function between LR and HR images can be learned by various methods, such as Gaussian process regression [77], kernel regression [78], weighted average [79], support vector regression [80] and sparse dictionary representation [26, 78, 81]. Here, sparse coding is a popular and effective methods, which can sparsely represent any natural image in a transform domain. It can effectively constrain the ill-posed problem of SR by embedding the prior knowledge while it is computationally expensive due to nonlinear reconstruction.

Recently, another category of neural networks based SR algorithms have been developed quickly in single-image SR field, especially for large image database such as ImageNet [82]. For these approaches, back-propagation algorithm [83] is generally employed to learn mapping function between LR and HR images. For example, inspired by sparse coding methods, Dong et al. [27] proposed a multiple-layer convolutional neural network (CNN) which directly learns end-to-end mapping function from LR images to HR images. With the development of CNN, there is a great improvement in algorithm efficiency, especially in memory cost and computational cost [84]. This advantage is quite important for real-time cell image SR in a lensless blood cell counting system because such a system requires limited usage of processing resources and enough system throughput, which will be introduced detailedly in Section 4.2

2.2 Machine Learning Basic

Machine learning is a kind of artificial intelligence (AI) that performs data analytics by automatically building model. It can enable computers to learn by themselves without being programmed explicitly. Because of the iterative aspect of machine learning, the trained model can adapt independently when exposed to new data. Therefore, it can perform high-quality data analytics and prediction in real time without human intervention. In this thesis, two kinds of machine learning algorithms, extreme learning machine and convolutional neural netowrk, will be introduced in terms of structure and training process.

2.2.1 Extreme Learning Machine

Extreme learning machine (ELM) is a novel machine learning algorithm based on single layer forward network (SLFN). As shown in Figure 2.4, this network is composed by three layers: input layer, hidden layer and output layer. The hidden node parameters, including input weight and bias, are generated randomly. And the output weight can be determined through least-square solution, which is described by Huang al. [85] in detail.

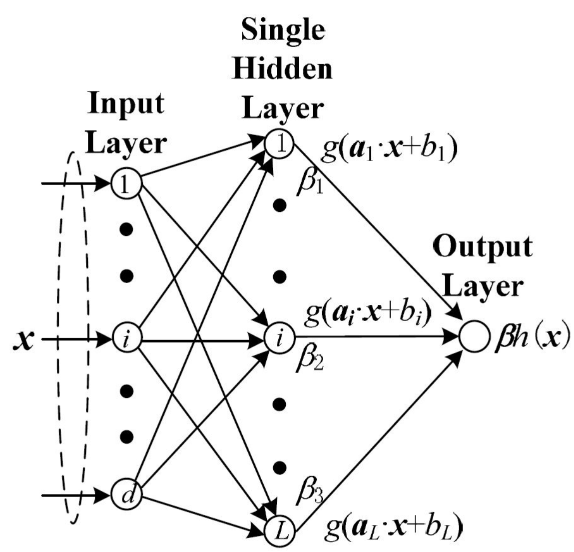

Figure 2.4: The structure of extreme learning machine (ELM) model

Given N arbitrary distinct training samples $\{(\mathbf{x}_i, \mathbf{y}_i)|\mathbf{x}_i \in \mathbf{R}^n, \mathbf{y}_i \in \mathbf{R}^m, i = 1, 2, ..., N\}$, the output function of standard ELM model with L hidden nodes can be modeled as:

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i G(\alpha_i, b_i, \mathbf{x}_j) = \mathbf{y}_j, j = 1, 2, ..., N \tag{2.1}$$

For SLFNs with additive hidden nodes, the output function can be expressed as:

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i g(\alpha_i \cdot \mathbf{x}_j + b_i) = \mathbf{y}_j, j = 1, 2, ..., N \tag{2.2}$$

where $\mathbf{x}_j = [x_{j1}, x_{j2}, ..., x_{jn}]^T$ and $\mathbf{y}_j = [y_{j1}, y_{j2}, ..., y_{jn}]^T$. α_i is the input weight vector connecting input neurons and the i^{th} hidden node. b_i is the bias of the i^{th}th hidden node. Both α_i and b_i are assigned randomly with independence of training samples.

Above formula can be shown into matrix form as follows

$$\mathbf{H}\beta = \mathbf{Y} \tag{2.3}$$

Here, $\mathbf{H} \in R^{N \times L}$ is the hidden layer output matrix.

By minimizing $||\mathbf{H}\beta - \mathbf{Y}||_2$, the output weight β can be calculated in training process. With least-squares solution, the corresponding estimation of β can be given by:

$$\beta = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Y} \tag{2.4}$$

2.2.2 Convolutional Neural Network

As a type of machine learning algorithm, convolutional neural network (CNN) is a feed-forward artificial neural networks composed by one or more convolutional layers [86]. Compared with ordinary neural networks with fully connected structure, CNNs has local-connection architecture that neurons of one layer are only connected to a small region of the layer before it, leading to easier training and fewer parameters [86, 87]. A typical architecture of CNNs includes convolutional layer, pooling layer and fully-connected layer. Taking a simple architecture (Input - Convolutional layer - ReLU layer - Pooling layer - Fully-connected layer) as an example [87], CNNs will be described as follows:

- Input: Input is the raw pixel values $h \times w \times d$ of a image. Here, h is the height of image, w refers to the width of image and w represents the number of channels. For example, an input $32 \times 32 \times 3$ means that the height and width are both 32 and there are three color channels R, G, B.

- Convolutional Layer: There are n filters with spatial size $f \times f$ in convolutional layer. These filters are convolved across the height and width of input to compute dot products. Composed by the dot products between filters and input at every spatial position, two-dimensional activation maps can be generated.

- ReLU Layer: The ReLU layer applies an elementwise activation function $max(0, x)$ to these filter response thresholding at zero.

- Pooling Layer: Each map generated in convolutional layer will be down-sampled with max or mean pooling operation along the spatial dimensions. Through reducing the spatial size of the input and decreasing the amount of parameters, this layer is helpful to reduce the network computation complex.

- Fully-connected Layer: The neurons in this layer is fully connected to all the nodes of previous layer, which is the same as the regular layer in ordinary neural network.

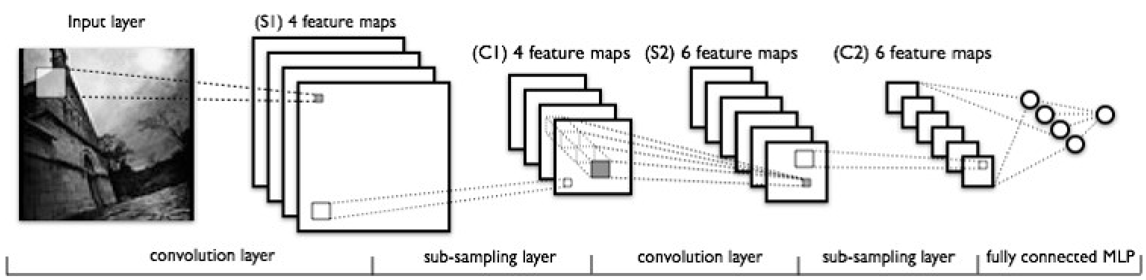

Figure 2.5: The structure of convolutional neural network

Figure 2.5 is a classic image showing the structure of CNN which is composed by two convolutional layers, two pooling layers and a fully-connected layer [88]. Moreover, back-propagation is generally utilized for training parameters of CNNs. Deploying proper loss function, parameters in convolutional layers and fully-connected layers can be determined by matching historical data to desired outputs in the training section.

Chapter 3

Smart Building Energy Management System

3.1 Building Energy Management System

This thesis proposed an IoT system for energy monitoring, planning and controlling for residential buildings. The building energy managment system (BEMS) is an ambient intelligence system that utilizes modern automation technology and advanced sensors to react towards residents' best experience by collecting environmental data and occupants' activity data. There are two sub-systems based on the BEMS architecture: indoor positioning system (IPS) and home automation system (HAS). Here, IPS can provide information about occupants' presence and movement in the building, which is indispensable to ensure high-quality performance of BEMS. And HAS can automatically turn on/off household appliance according to temperature, humidity, light intensity and occupants activity information. For example, household appliances, such as air-conditioner, humidifier and light, can be powered automatically and set in a comfort model when occupants enter the room. And when nobody is in the room, all the appliances will be turned off for energy saving, except security system and fridge. Moreover, during the peak load period, the stored solar energy is scheduled to be supplied to the building for alleviating the great demand pressure of main electricity power-supply.

The overall BEMS with distributed architecture can be shown as Figure 3.1. The system infrastructures are implemented in each room on the base of smart gateway network for capturing and analyzing data [89]. Since data-analyzing and decision-making are performed in each smart gateway independently, this distributed architecture enables system work in a more stable way. Even if one gateway at room level is broken, the function of overall system will not be influenced. The main components of BEMS include computation unit, sensor unit, communication unit and power-grid.

Computation Unit Smart gateway is the central computation unit for intense computation and control. It can link smart devices, smart meters and smart appliances in the local network. Also, it can continuously collect energy consumption data and environmental data from smart meters, and remotely control smart household appliances according to data analytics results.

In this thesis, BeagleBoard-xM with AM37x1GHz ARM, an open hardware, is selected as the smart gateway. This board is designed with an improvement in the power level module and storage module. With its four USB ports, many extensions can be achieved on this board. For example, the ZigBee extension can be used for connecting household appliances with the computation unit in a fast way. Moreover, a Linux based Ubuntu system is embedded into the BeagleBoard-xM as the operating system. Based on this embedded operating system, various IoT applications can be freely and efficiently developed due to several advantages. First, many suppliers support software development with no royalties or licensing fee. Also, this operating system has a stable kernel with ability to read, modify and redistribute source code [90]. With the BeagleBoard-xM as the computation center and basic platform, both household appliance control and occupants' position monitoring can be performed.

Sensor Unit Various sensors are deployed for collecting energy data, environmental data and movement data. Main sensors in BEMS are listed as follows:

- **Environmental Sensors:** Environmental sensors include temperature sensors, humidity sensors and light intensity sensors. They can be used to collect environmental factors and provide the information to computation unit for analyzing.

- **Smart Power-meter:** Smart power-meter is a kind of energy sensor. It not only can sense current in real time, also can record energy consumption data. Moreover, it can change the supplied energy source physically utilizing an included switch.

- **Wireless Adapter:** Wireless adapter is a Wi-Fi sensor. It can be used to capture Wi-Fi packets emitted from mobile devices according to IEEE 802.11

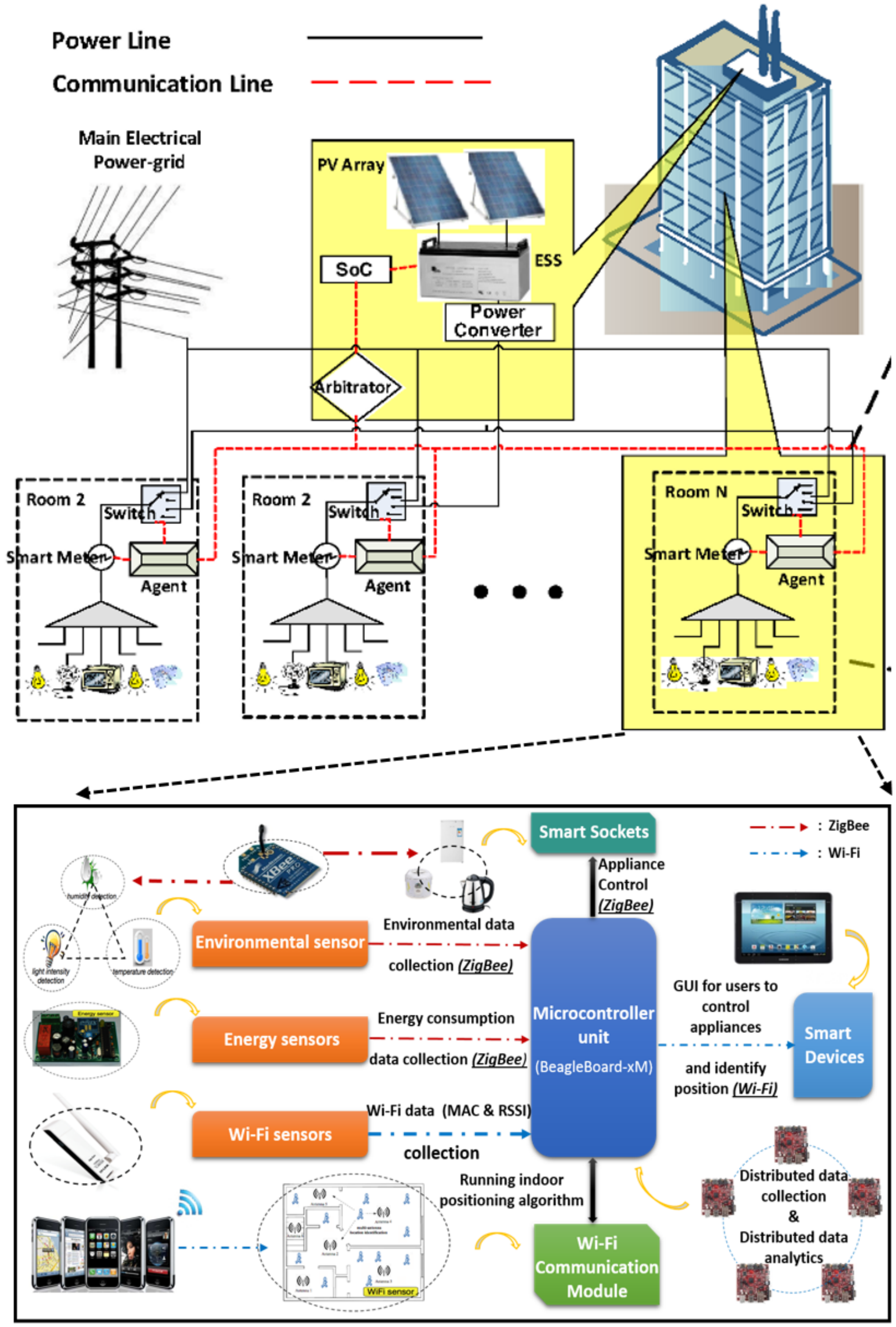

Figure 3.1: The overview of smart building energy management system

protocol. These captured packets are stored, parsed and computed in computation unit for positioning.

Communication Unit The proposed IoT system is designed and built based on wireless network because it has advantage of low cost and low power usage. Internet protocol (IP) can perform data deliver from source to destination according to an IP address and perform data communication within one packet-switched internetwork. As a popular IP-based communication technology, Wi-Fi can be setup and adopted widely. Almost all the electronic devices, such as mobile phones, laptops, personal computers and peripheral devices, have the function to access the network via Wi-Fi. Moreover, since Wi-Fi can be setup easily without extra personal tag and can be accessed with low cost, the sensor network based Wi-Fi can overcome main challenges faced by other indoor sensor network, such as complex configuration, time-consuming deployment and expensive operations.

In the proposed system, the Wi-Fi packet is captured by the TL-WN722N wireless adapter according to IEEE 802.11 protocol. The captured Wi-Fi packet contains one important information, Management and Control Address (MAC), in a 30-byte-length header, and another significant information, received signal strength indicator (RSSI), in the radio-tap header. Radio-tap header and Wi-Fi header are shown in Figure 3.2. With the MAC address to identify objects and RSSI values to reflect distance information, the Wi-Fi signal can also be used for position monitoring.

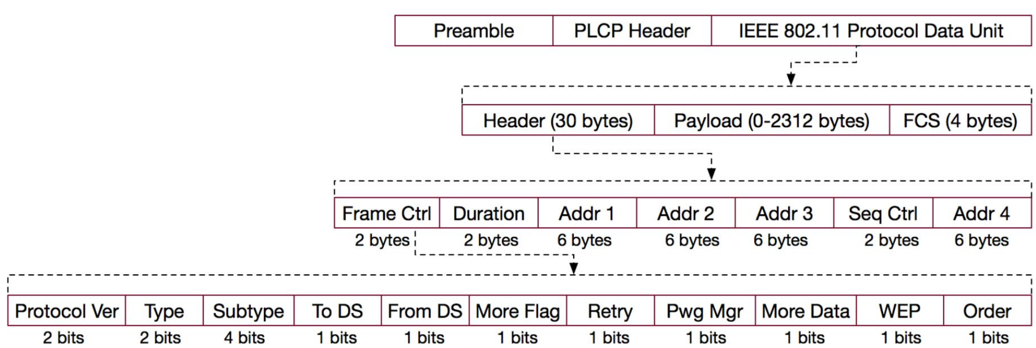

Figure 3.2: Frame format in Wi-Fi header field

Power-grid Smart building is supplied with energy by both the main electricity power-grid and the additional power-grid of renewable energy.

- **Main electricity power-grid:** As an external electricity supplier, the main power-grid is the primary energy supplier for residential buildings, whose price is much higher compared to renewable energy supplier.
- **Additional power-grid of solar energy:** Additional power-grid is the solar photovoltaic constructed by connected photovaltaic cells [91]. The solar energy can be stored in the energy storage module [92] and supplied to customers during the peak load period.

Graphical User Interface Smart devices with graphical user interface (GUI) are designed for providing information to users in a visual interface, which can satisfy the requirements about user-friendly and simple integration of numerous networked appliances [93]. Occupants can not only learn about historical or real-time power consumption as well as environmental factors record, but also can remotely control household appliances through the smart phone or tablet.

3.2 Distributed Machine Learning based Data Analytics

3.2.1 Distribution of Online Sequential ELM

Review of OS-ELM

To provide sufficient background for developing distribution of online sequential extreme learning machine (DOS-ELM), this section briefly introduces related work of online sequential extreme learning machine (OS-ELM) proposed by Liang et al. [94,95]. OS-ELM is developed on the basis of ELM for single layer feedforward networks (SLFNs) with additive and radial basis function (RBF), which can learn data chunk-by-chunk or one-by-one.

There are mainly two phases of OS-ELM algorithm: initialization phase and sequential learning phase.

Initialization phase: N_0 training samples $\Gamma = \{(\mathbf{x}_i, \mathbf{y}_i)\}_i^{N_0}$ are used to initialize the learning model. Here, N_0 is required to be equal or greater than the number of hidden node L, i.e. $N_0 \geqslant L$, in order to guarantee OS-ELM perform the same learning performance as original ELM with $rank(\mathbf{H}_0) = L$.

- Set k as the number of chunks of training data which is input to the network. In the initialization phase, k is set as zero.

- Generate input weight parameter α_i and bias b_i randomly.

- Calculate the initial output matrix $\mathbf{H}_0$ with L hidden nodes:

$$\begin{bmatrix} G(\alpha_1, b_1, \mathbf{x}_1) & \cdots & G(\alpha_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\alpha_1, b_1, \mathbf{x}_1) & \cdots & G(\alpha_L, b_L, \mathbf{x}_{N_0}) \end{bmatrix}_{N_0 \times L}$$

- Calculate the initial output weight $\beta^{(0)}$:

$$\beta^{(0)} = (\mathbf{H}_0^T \mathbf{H}_0)^{-1} \mathbf{H}_0^T \mathbf{Y}_0 = \mathbf{P}_0 \mathbf{H}_0^T \mathbf{Y}_0 \tag{3.1}$$

  where

$$\begin{aligned} \mathbf{P}_0 &= (\mathbf{H}_0^T \mathbf{H}_0)^{-1} \\ \mathbf{Q}_0 &= \mathbf{P}_0^{-1} = \mathbf{H}_0^T \mathbf{H}_0 \end{aligned} \tag{3.2}$$

Sequential learning phase The $(k+1)^{th}$ chunk of training data $\Gamma = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=(\sum_{j=0}^{k} N_j)+1}^{\sum_{j=0}^{k+1} N_j}$ is input to the network. Here N_{k+1} is referred as the number of the training sample in the $(k+1)^{th}$ chunk.

- **(a)** Compute the partial output matrix $\mathbf{H}_{k+1}$ with L hidden node:

$$\begin{bmatrix} G(\alpha_1, b_1, \mathbf{x}_{(\sum_{j=0}^{k} N_j)+1}) & \cdots & G(\alpha_L, b_L, \mathbf{x}_{(\sum_{j=0}^{k} N_j)+1}) \\ \vdots & \ddots & \vdots \\ G(\alpha_1, b_1, \mathbf{x}_{\sum_{j=0}^{k+1} N_j}) & \cdots & G(\alpha_L, b_L, \mathbf{x}_{\sum_{j=0}^{k+1} N_j}) \end{bmatrix}_{N_{k+1} \times L}$$

- **(b)** Compute the output weight for the $(k+1)^{th}$ chunk:

  With $\mathbf{Y}_{k+1} = [\mathbf{y}_{\sum_{j=0}^{k} N_j)+1}, ..., \mathbf{y}_{\sum_{j=0}^{k+1} N_j}]^T_{N_{k+1} \times m}$, the output weight $\beta^{(k+1)}$ can be calculated as follows:

$$\beta^{(k+1)} = \beta^{(k)} + \mathbf{Q}_{k+1}^{-1} \mathbf{H}_{k+1}^T (\mathbf{T}_{k+1} - \mathbf{H}_{k+1} \beta^{(k)}) \tag{3.3}$$

  where

$$\mathbf{Q}_{k+1} = \mathbf{Q}_k + \mathbf{H}_{k+1}^T \mathbf{H}_{k+1} \tag{3.4}$$

Since $\beta^{(k+1)}$ can be calculated using $\mathbf{Q}_{k+1}^{-1}$, Woodbury formula [96] is applied to modify the equation for avoiding the inverse calculation in the recursive process.

$$\begin{aligned}\mathbf{Q}_{k+1}^{-1} &= (\mathbf{Q}_k + \mathbf{H}_{k+1}^T\mathbf{H}_{k+1})^{-1} \\ &= \mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1}\mathbf{H}_{k+1}^T(\mathbf{I} + \mathbf{H}_{k+1}\mathbf{Q}_k^{-1}\mathbf{H}_{k+1}^T)^{-1}\mathbf{H}_{k+1}\mathbf{Q}_k^{-1}\end{aligned} \tag{3.5}$$

With $\mathbf{P}_{k+1} = \mathbf{Q}_{k+1}^{-1}$, the $\beta^{(k+1)}$ can also be expressed as:

$$\beta^{(k+1)} = \beta^{(k)} + \mathbf{P}_{k+1}\mathbf{H}_{k+1}^T(\mathbf{T}_{k+1} - \mathbf{H}_{k+1}\beta^{(k)}) \tag{3.6}$$

where

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{P}_k\mathbf{H}_{k+1}^T(\mathbf{I} + \mathbf{H}_{k+1}\mathbf{P}_k\mathbf{H}_{k+1}^T)^{-1}\mathbf{H}_{k+1}\mathbf{P}_k \tag{3.7}$$

- **(c)** Assign $k = k + 1$ and go to **(a)** in sequential learning phase to process next chunk of training data.

OS-ELM with Incremental Least-Square Solver

The number of hidden node is initially assigned as a small integer number in application. With the number of training samples increasing, the training accuracy may become worse due to the limitation of hidden node number. Therefore, it is necessary to increase the number of hidden node to reduce the training error. Since large number of hidden node will cause increasing of computational cost and time consuming, an incremental solution is needed for hidden node number adjustment with low computation complexity.

Cholesky decomposition with computational cost $O(\frac{1}{3}L^3)$ is adopted to solve the least square problem of minimizing $||\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}||_2$. Compared with SVD and QR decomposition with computational cost $O(4NL^2 - \frac{4}{3})$ and $O(2NL^2 - \frac{2}{3}L^3)$ respectively, its incremental and symmetric property saves half memory and reduces the computational cost [97].

Here, $\mathbf{H}_L$ is denoted as the matrix with L number of hidden neuron nodes ($L < N$). We decompose the symmetric positive definite matrix $\mathbf{H}^T\mathbf{H}$ into

$$\mathbf{H}_L^T\mathbf{H}_L = \mathbf{G}_L\mathbf{G}_L^T \tag{3.8}$$

where T is transpose operation of the matrix and $\mathbf{G}_L$ represents a low triangular matrix.

$$\begin{aligned}\mathbf{H}_L^T\mathbf{H}_L &= \left[\mathbf{H}_{L-1}\ h_L\right]^T\left[\mathbf{H}_{L-1}\ h_L\right] \\ &= \begin{pmatrix}\mathbf{H}_{L-1}^T\mathbf{H}_{L-1} & \mathbf{p}_L \\ \mathbf{p}_L^T & s\end{pmatrix}\end{aligned} \tag{3.9}$$

where h_L is the new column added by increasing the number of hidden nodes L and can be calculated from (2.1).

The Cholesky matrix can be expressed as

$$\begin{aligned}&\mathbf{G}_L\mathbf{G}_L^T \\ &= \begin{pmatrix}\mathbf{G}_{L-1} & \mathbf{0} \\ \mathbf{q}_L^T & t\end{pmatrix}\begin{pmatrix}\mathbf{G}_{L-1}^T & \mathbf{q}_L \\ \mathbf{0} & t\end{pmatrix}\end{aligned} \tag{3.10}$$

According to Equation 3.8, we can easily calculate the $\mathbf{q}_L$ and scalar t for Cholesky factorization as

$$\mathbf{G}_{L-1}\mathbf{q}_L = \mathbf{p}_L,\ t = \sqrt{s - \mathbf{q}_L^T\mathbf{q}_L} \tag{3.11}$$

where $\mathbf{G}_{L-1}$ is the previous Cholesky decomposition result and $\mathbf{p}_L$ can be known from (3.9). When $l = 1$, $\mathbf{G_1}$ is a scaler and equals with $\sqrt{\mathbf{H_1}^T\mathbf{H_1}}$. In this way, we can continue using previous factorization result and update only according part.

Distribution of OS-ELM

DOS-ELM is composed by several parallel OS-ELM networks with the same activation function and the same number of hidden nodes. For each SLFN in DOS-ELM, input parameters are assigned randomly and the output weight is computed based on sequentially new arriving training data. After paralleling operation of each network, the final result is determined by the average of outputs of all the independent OS-ELM networks.

The architecture and working flow of DOS-ELM can be described in Figure 3.3. N_0 original training samples are sent to each gateway for initial training process independently and then new arriving data is input to each gateway for sequential training in parallel operation pattern. The trained weight of each network is stored in a small data storage for sequential learning process. In the process of real-time application,

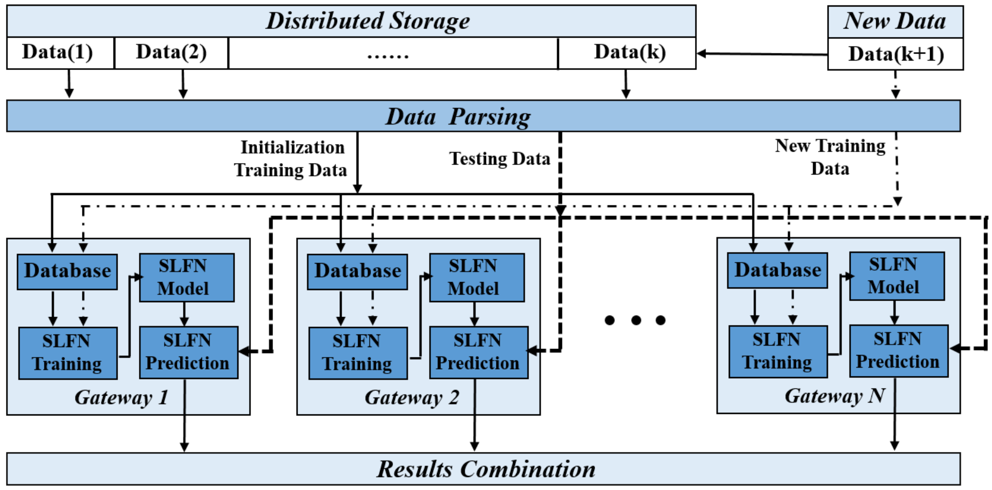

Figure 3.3: Architecture of DOS-ELM

each well-trained network is used to analyze the same format input data and generate prediction result independently. To determine the final output, all the results from each OS-ELM network are collected using message passing interface (MPI) protocol and computed in the central gateway. Considering the result of each network is $r^{(i)}(\mathbf{x}_i)$, $j = 1, 2, ..., M$, the final output can be expressed as follows:

$$r(\mathbf{x}_i) = \frac{1}{M} \sum_{M}^{j=1} r^{(j)}(\mathbf{x_i}) \tag{3.12}$$

here, M is the number of distributed OS-ELM network and $r(\mathbf{x}_i)$ is the final output with $\mathbf{x}_i$ as input.

In this distribution system, each OS-ELM network is distinct due to randomly and independently generated parameters. When the new arriving data is input into each distinct network, some of them can adapt well and fast while some may not. Since the DOS-ELM is regarded as a whole system, it can keep a good prediction result even if individual network may not adapt well to the new arriving data. In other words, the output of the whole DOS-ELM system fluctuates in a smaller range in comparison with the output of single OS-ELM network. When the number of single OS-ELM network is getting larger, the prediction result of the whole system is getting more accurate. Moreover, with the number of OS-ELM network increasing, the training time and testing time will not increase obviously because each OS-ELM network works in parallel.

3.2.2 Occupant Behavior Profile

Occupant Behavior Definition

In this thesis, occupant behavior is represented by the information about occupants' movement combining with the function of specific room. Here, the information about occupants movement mainly focuses on whether the room is occupied, how many occupants are in the room, what time the occupants enter/leave the room and how long occupants present. Since the function of each room is different with various kinds of household appliances, occupant behavior in each room is of quite typical features with different energy consumption levels. For instance, when occupants enter kitchen and move around actively, it is highly possible for occupant to use microwave, rice cooker, kettle and other household appliances related to cooking, resulting in high energy consumption level. When the occupant is present in the living room, it is possible for him to watch TV, surfing the Internet on personal computer or carry on other activities related to entertainment, which will lead to low level of energy consumption. Therefore, we can classify occupant behavior into several types according to the function of room, and then detect occupant behavior through indoor positioning information. For residential buildings, main rooms generally includes kitchen, bedroom, living room and washing room. Considering general types of household appliances, the occupant behavior can be generally categorized as shown in Table 3.1.

Occupant Behavior Cognition

Occupants' movement and presence can be monitored by real-time indoor positioning system through Wi-Fi signal analytics as shown in Figure 3.4. The system is composed by a transmitter unit and a measuring unit. Here, smart mobile devices emit Wi-Fi signal as transmitter unit. Smart gateway (BeagleBoard-xM) works as the measuring unit to collect Wi-Fi signal. As the central unit of the whole system, the BeagleBoard-xM can also store captured data, parse Wi-Fi packets and compute positioning algorithm.

Wi-Fi packets, captured by Wi-Fi adapter (TL-WN722N) in monitor mode, includes two important information for indoor positioning. One is media access control address (MAC) that is unique for each mobile device. It can be used to identify the

Table 3.1: Relationship Between Room Type and Occupants Behavior

Room Type		Kitchen	Bedroom/Living room	Washroom
Behavior		*Cooking/Eating*	*Rest/Entertainment*	*Washing/Cleaning*
Functional Appliances	Microwave	✓		
	Rice cooker	✓		
	Kettle	✓		
	Dish washer	✓		
	Fridge	✓		
	Toaster	✓		
	TV		✓	
	Personal computer		✓	
	Laptop		✓	
	Washing machine			✓
	Cloth dryer			✓
	Water heater			✓
	Bath heater			✓
General Appliances	Lights	✓	✓	✓
	Heating		✓	✓
	Air-conditioning		✓	
	Ventilation	✓		✓

device that emits the Wi-Fi packet. Another information is received signal strength indicator (RSSI) that is a reflection of distance between the transmitter unit and the measuring unit. Considering captured MAC address and RSSI value of each Wi-Fi packet, DOS-ELM can be trained on smart gateways. The trained weights for networks are stored and then used for real-time positioning application with same format data as input. The positioning prediction result of each smart gateway will be collected through MPI protocols and processed using soft-voting to determine the final positioning result.

Occupant Behavior Extraction

To extract occupant behavior according to occupant positioning information, positioning record should be organized as training data format. Since the occupant behavior is different in each room with specific function, the detected occupant positioning record should be processed respectively.

Figure 3.4: Indoor Positioning System Overview

For each room r, occupant positioning can be categorized into four states represented by S:

$$S = \begin{cases} s_1 : 0 & \text{no occupant in the room } r \\ s_2 : 1 & \text{occupants in the room } r \\ s_3 : 0 \longrightarrow 1 & \text{occupants entering the room } r \\ s_4 : 1 \longrightarrow 0 & \text{occupants leaving the room } r \end{cases} \tag{3.13}$$

here, occupant positioning state S is monitored by real-time indoor positioning system through Wi-Fi signal detection and analytics. 1 represents occupants' presence in the room while 0 indicates the room is not occupied.

The probability that occupants are present in the room r can be computed as follows:

$$b(i) = \frac{T(s_2) + T(s_3)}{T_d}, i = 1, 2, ..., 24 \tag{3.14}$$

here, $b(i)$ represents the occupants movement probability of room r in the i^{th} time interval. Here, the length of time interval T_d is set as 60 minutes, which results in 24 time intervals. $T(s_2)$ and $T(s_3)$ are the time length when occupants are present in the room r.

For the room r, the daily occupant movement probability $\mathbf{B}$ can be represented based on $b(i)$:

$$\mathbf{B} = [b(1), b(2), b(3), ..., b(24)]^T \tag{3.15}$$

where, the daily occupant movement probability $\mathbf{B}$ is referred as the occupant behavior profile.

For the dth day, the occupant behavior profile $\mathbf{B}_d$ can be predicted according to the trend of behavior profile in previous several days. Here, we choose previous seven-day occupant behavior profiles as the training data $\mathbf{X}_B$ and the 8th-day behavior profile as the training data $\mathbf{Y}_B$, which can be shown as follows:

$$\begin{aligned} \mathbf{X}_B &= \{\gamma_d \cdot \sigma \cdot \mathbf{B}_d | 1 \leqslant d \leqslant 7\} \\ \mathbf{Y}_B &= \{\mathbf{B}_d | d = 8\} \end{aligned} \tag{3.16}$$

here, $\mathbf{B}_d$ is the occupant behavior profile on the d^{th} day.

γ_d is the daily weight parameter which should be assigned as:

$$\gamma_{d-1} > \gamma_{d-2} > ... > \gamma_{d-7} \tag{3.17}$$

here, since the latest captured data has greater effect on prediction result, the new arriving data should be paid more attention by assigning heavy weight.

σ is the weaken factor used to reduce the effect of abnormal value, which can be computed as:

$$\begin{aligned} &\sigma = \begin{cases} 1 & \frac{1}{2}A \leqslant \bar{\mathbf{B}} \leqslant \frac{3}{2}A \\ 0.1 & \bar{\mathbf{B}} < \frac{1}{2}A \ \text{ or } \bar{\mathbf{B}} > \frac{3}{2}A \end{cases} \\ &s.t \\ &\bar{\mathbf{B}} = \frac{\sum_{i=1}^{24} b(i)}{24} \qquad A = \frac{\sum_{d=1}^{7} \bar{\mathbf{B}}_\mathbf{d}}{7} \end{aligned} \tag{3.18}$$

here, these extremely abnormal data can be detected using the average of movement probability value in previous days. The upper bound is set as $\dfrac{3}{2}A$ and the lower bound is set as $\dfrac{1}{2}A$.

3.2.3 Energy Consumption Profile

Energy consumption profile E is composed by hourly energy consumption $e(t)$ in the time span of 24 hours, which can be expressed as:

$$\mathbf{E} = [e(1), e(2), ..., e(24)] \tag{3.19}$$

The characteristic of daily energy consumption profile is various due to different outside weather, inside environmental factors, day types as well as occupant behavior profiles. The input features are vital for the accuracy of prediction model.

Table 3.2: Input Feature For Daily Load Forecasting

Inputs	**Descriptions**
1-24	H(d-1,t), H(d-2,t), H(d-3,t), H(d-4,t), H(d-5,t), H(d-6,t), H(d-7,t): Humidity of the 7 days before the predicted day at the same time
25-48	T(d-1,t), T(d-2,t), T(d-3,t), T(d-4,t), T(d-5,t), T(d-6,t), T(d-7,t): Temperature of the 7 days before the predicted day at the same time
49-72	L(d-1,t), L(d-2,t), L(d-3,t), L(d-4,t), L(d-5,t), L(d-6,t), L(d-7,t): Light intensity of the 7 days before the predicted day at the same time
73	Date type: weekend is expressed as 0 and weekday is represented by 1

To forecast the daily energy consumption on the $(d + 1)^{th}$ day, the actual data, including temperature, humidity and light intensity, of previous days and the type of $(d + 1)$th day are referred as training features shown as Table 3.2:

3.3 Solar Energy Allocation for Peak Load Reduction

During the peak load period, the energy demand is possible to exceed the upper limit of main electricity power-grid supply level, which will lead to load shedding and power outage. Since the electricity price is adjusted dynamically in accordance with the fluctuation of energy demand, the price of main electricity power-grid is usually higher during the peak load period. In order to compensate the peak load demand of the main electricity power-grid and save the energy cost for users, a dynamic solar energy allocation strategy has been proposed. Based on the predicted energy consumption profile and occupant behavior profile, the peak load period can be detected and the solar energy can be allocated for each room as an additional power supply.

3.3.1 Problem Formulation

For each hour t, the generated solar energy is denoted as $S_{Generation}(t)$ and the energy demand of the main electricity power-grid is referred as $E_{Demand}(t)$. The solar energy allocation amount $S_{Allocation}(t)$ can be determined by predicted occupant behavior profile and energy consumption profile:

$$S_{Allocation}(t) = \mathbf{f}(\mathbf{B}, \mathbf{E}) \tag{3.20}$$

here, **B** is the predicted occupant behavior profile. **E** is the energy consumption profile.

Two objectives of solar energy allocation strategy are defined:

Objective 1: Aiming at peak load reduction, the solar energy allocation strategy should minimize the standard deviation of actual energy consumption from main electricity power-grid, which can be expressed as:

$$argmin(dev \sum_{t=1}^{t=24} (E_{Demand}(t) - S_{Allocation}(t))) \tag{3.21}$$

Objective 2: The solar energy allocation strategy is expected to minimize the daily electricity cost, which can be shown as:

$$argmin \sum_{t=1}^{t=24} (E_{Demand}(t) - S_{Allocation}(t))p(t) \tag{3.22}$$

here, $p(t)$ is the real-time electricity price of main electricity power-grid.

Note that the solar energy allocation amount $S_{Allocation}(t)$ can not exceed the available amount of solar energy storage at anytime. This constraint can be expressed as:

$$S_{Allocation}(t) \leqslant S_{Storage}(t) \tag{3.23}$$

here, $S_{Storage}(t)$ is the available amount of solar energy at time t, which is stored in the energy storage module of the additional power-grid.

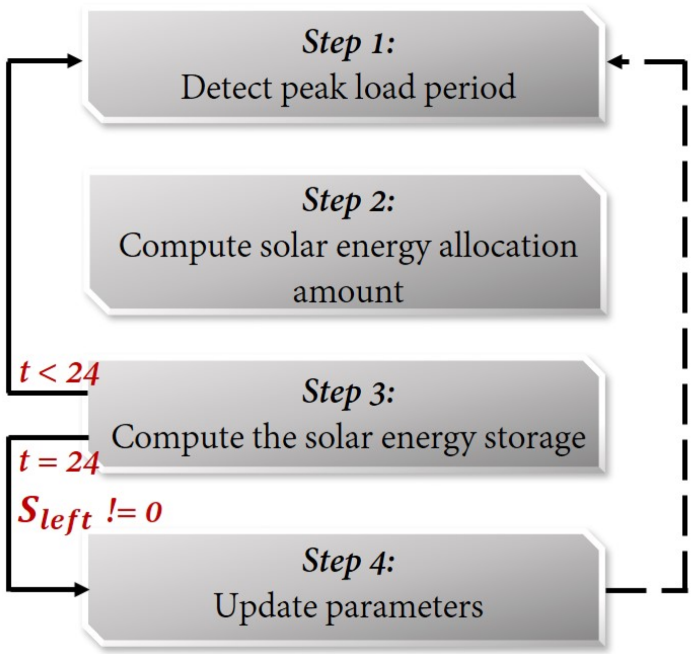

Figure 3.5: The flow of solar energy allocation

3.3.2 Solar Energy Allocation Strategy

To realize above two objectives and achieve the optimal utilization of solar energy, a dynamic solar energy allocation strategy has been proposed, which can be summarized as shown in Figure 3.5:

Step 1: Peak Load Period Detection To formally illustrate the parameter used for peak load detection, two factors will be defined firstly.

(1) $P_B(t)$ is defined as the occupants movement probability for the whole house at time t:

$$P_B(t) = \frac{\sum_{i=1}^{M} \theta_i b_i(t)}{M}, \quad 0 \leqslant \theta_i \leqslant 1 \tag{3.24}$$

here, the $b_i(t)$ is the components of occupant behavior profile $\mathbf{B}$, which represents the occupant movement probability for room i at time t. For each room with specific function, household appliances are diverse leading to different energy consumption levels. To identify the level of energy consumption in each room, weight parameter θ_i is assigned. And M represents the number of rooms in the house.

(2) $P_E(t)$ is referred as the proportion of energy consumption at time t:

$$P_E(t) = \frac{e(t)}{\sum_{t=1}^{24} e(t)} \tag{3.25}$$

here, $e(t)$ represents the component of predicted 24-hour energy consumption profile $\mathbf{E}$.

When occupants are present and move around actively in the room, energy consumption is quite likely to increase [98]. Therefore, the peak load period is detected by fusing the probability of occupant movement $P_B(t)$ and the proportion of energy consumption $P_E(t)$. The probability of peak load $P_{peak}(t)$ with weight parameter η can be expressed as:

$$P_{\text{peak}}(t) = \eta P_B(t) + (1-\eta)P_E(t), 0 \leqslant \eta \leqslant 1 \tag{3.26}$$

With a pre-assigned upper limitation, the peak load moment can be detected when $P_{peak}(t)$ exceeds this upper limitation. At such moment, a specific amount of solar energy should be allocated.

Step 2: Allocation Amount Computation During the peak load period, the expected amount of solar energy allocation $A_E(t)$ can be determined by

$$A_E(t) = \alpha(\mathbf{E(t)} - \bar{\mathbf{E}}) + \beta P_B(t) \tag{3.27}$$

here, $\bar{E}$ represents the average value of predicted 24-hour energy consumption. α and β are initialized according to actual energy demand. To perform optimal utilization of solar energy, α and β should be adjusted until the total amount of daily generated solar energy can be used up.

Since the amount of solar energy storage is regarded as a a limiting condition, the actual amount of solar energy allocation can be expressed as:

$$A(t) = \begin{cases} A_E(t) & S_{Storage}(t) \geqslant A_E(t) \text{ and } e(t) > A_E(t) \\ e(t) & S_{Storage}(t) \geqslant A_E(t) \text{ and } e(t) < A_E(t) \\ S_{Storage}(t) & S_{Storage}(t) < A_E(t) \text{ and } e(t) > S_{Storage}(t) \\ e(t) & S_{Storage}(t) < A_E(t) \text{ and } e(t) < S_{Storage}(t) \end{cases} \tag{3.28}$$

here, $S_{Storage}(t)$ is the total available amount of solar energy at time t. $e(t)$ is the energy demand at time t which is the component of energy consumption profile $\mathbf{E}$.

Step 3: Solar Energy Storage If the solar energy amount allocation is less than the total available amount of solar energy at time t, the remained solar energy $S_{Left}(t)$ can be computed in different cases:

$$S_{Left}(t) = \begin{cases} S_{Storage}(t) - A_E(t) & S_{Storage}(t) \geqslant A_E(t) \text{ and } e(t) > A_E(t) \\ S_{Storage}(t) - e(t) & S_{Storage}(t) \geqslant A_E(t) \text{ and } e(t) < A_E(t) \\ 0 & S_{Storage}(t) < A_E(t) \text{ and } e(t) > S_{Storage}(t) \\ S_{Storage}(t) - e(t) & S_{Storage}(t) < A_E(t) \text{ and } e(t) < S_{Storage}(t) \end{cases} \tag{3.29}$$

The remained solar energy at time t can be accumulated for next allocation period.

For next time $t+1$, the solar energy storage $S_{Storage}(t+1)$ can be determined by

$$S_{Storage}(t+1) = S_{Left}(t) + S_{Generation}(t) \tag{3.30}$$

Step 4: Parameter Adjustment For the optimal utilization of solar energy, the stored solar energy should be used up at the end of one day, which can be expressed as:

$$S_{Left}(24) = 0 \tag{3.31}$$

If there is remained solar energy at the end of one day, the parameter α and β should be adjusted to assign the expected amount $A_E(t)$ of solar energy allocation as a more proper value. Then go back to the $Step 2$.

The whole flow on smart-gateway platform in BEMS can shown in Figure 3.6

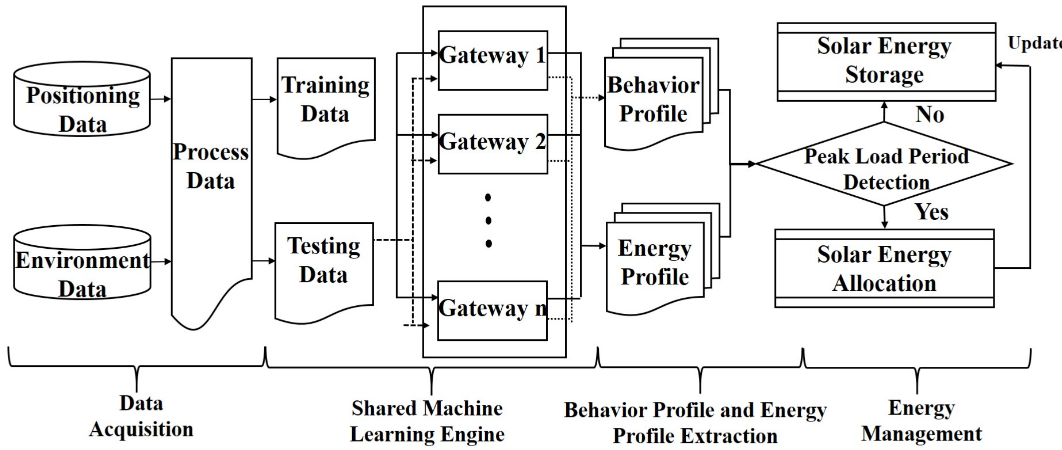

Figure 3.6: Flow of data analytics for building energy management system

3.4 Results and Discussion

3.4.1 Setup and Benchmark

In this section, daily occupant behavior profile and daily energy consumption profile are predicted through the proposed DOS-ELM method. In order to estimate the performance of DOS-ELM, support vector regression (SVR) is adopted to forecast occupants behavior profile and energy consumption profile for comparison. The prediction accuracy is estimated by mean absolute percentage error (MAPE) and root mean square error (RMSE). MAPE and RMSE can be defined as:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}|\frac{P(t)-R(t)}{R(t)}| \tag{3.32}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(P(t)-R(t))^2} \tag{3.33}$$

where P(t) is the prediction value and R(t) is the actual value.

Then, based on the prediction results of occupant behavior profile and energy consumption profile, peak load period can be detected and solar energy can be allocated to the smart building for reducing the peak load demand of main power-grid and also saving the daily energy cost.

In this experiment, one dataset from the Smart* Home Dataset [99] and a dataset

Table 3.3: Summary List of Experiment Set-up

Parameter	Value
Number of Gateway	5
Initial Training Days	7 Days
Continue Evaluation Days	23 Days
Environmental Data	30 Days
Occupant Motion Data	30 Days

provided by Energy Research Institute of Nanyang Technological University are utilized for short-term load forecasting. A real-time electricity pricing strategy from [100] is adopted to elaborated the performance of the proposed solar energy allocation strategy.

The data analytics and prediction based on DOS-ELM are running on the computing platform of BeagleBoard-xM with AM37x 1GHz ARM processor.

Experiment set-up is described in Table 3.3.

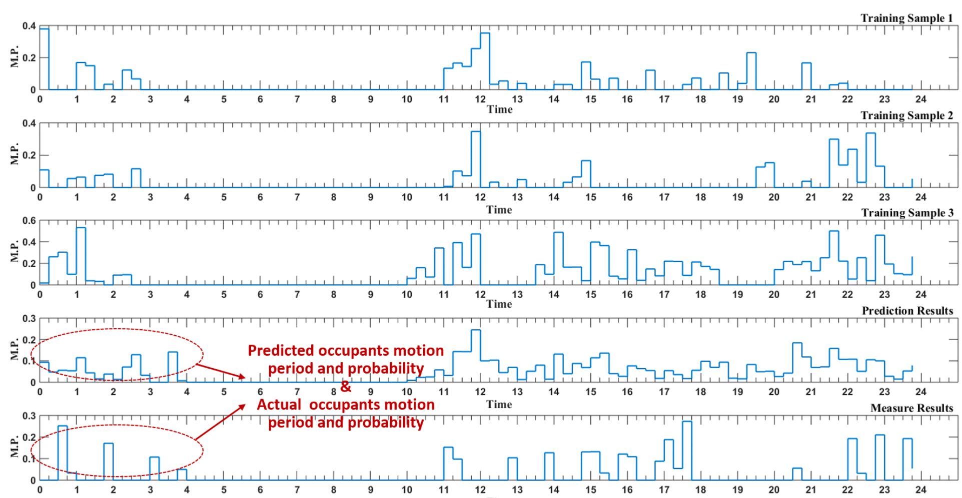

Figure 3.7: Motion probability within 15 minutes interval in a living room (3 training samples, 1 predicted result and 1 actual result).

3.4.2 Occupant Profile Extraction

Daily occupant behavior profile is predicted and represented by movement probability in fifteen-minute interval as shown in Figure 3.7. The measured actual data and corresponding training data are also displayed in the Figure 3.7 to verify the prediction accuracy. It is obvious that the predicted profile has the same trend as actual one, which means the predicted active period and movement probability is similar to the actual ones. This relatively accurate prediction result is quite helpful for energy management strategy which will be introduced in next section. In order to estimate the daily occupant movement profile for the whole residential house, occupant movement profiles in four rooms, including living room, kitchen, basement and bedroom, are predicted as shown in Figure 3.8 and combined for analytics.

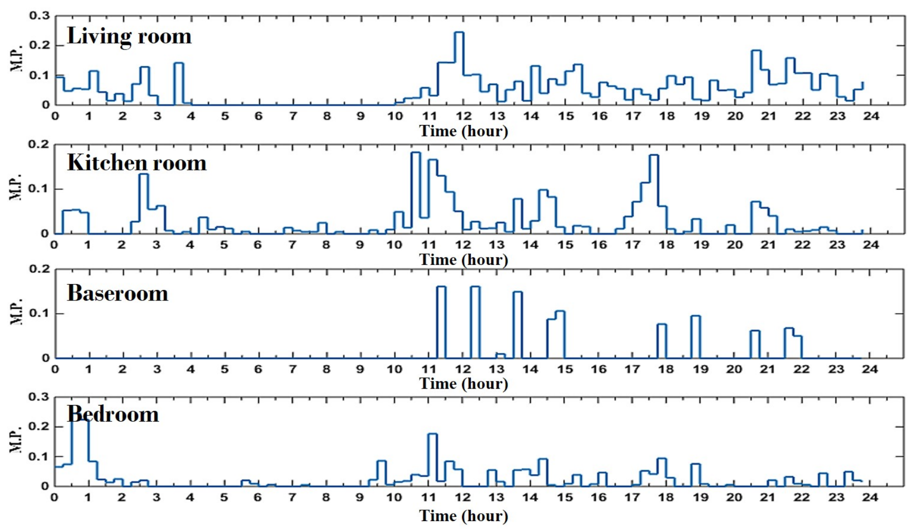

Figure 3.8: Motion probability within 15 minutes interval in a living room, kitchen, baseroom, bedroom.

3.4.3 Energy Profile Extraction

Daily energy consumption profile is extracted using proposed DOS-ELM and SVR respectively as shown in Figure 3.9. The energy consumption profile is various for different days while the proposed DOS-ELM can capture the peak load period and predict the load demand value accurately. For extremely high peak load, both DOS-ELM and SVR has no capability to predict the exact load value while the result of DOS-ELM is relatively more similar to the actual one.

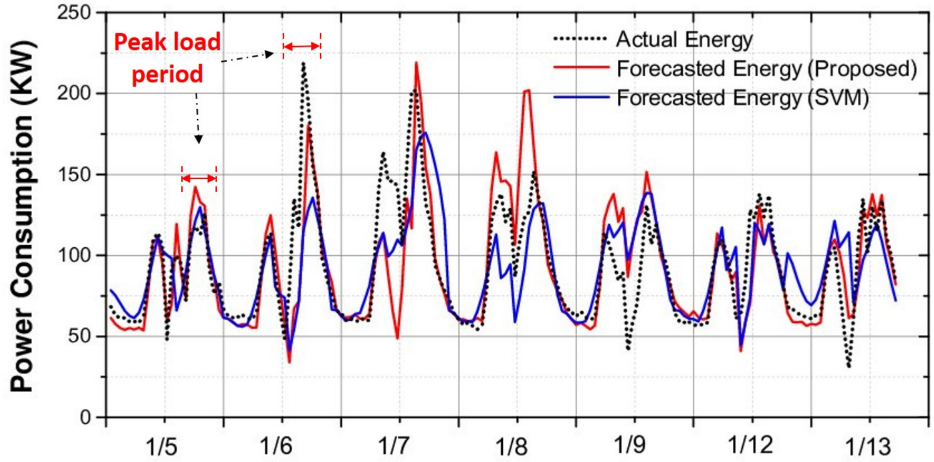

Figure 3.9: Daily energy forecasting using DOS-ELM and SVR respectively

The prediction accuracy is estimated using MAPE and RMSE as shown in Table 3.4. Comparing with the performance of SVR, the DOS-ELM can achieve better prediction by 14.60% and 14.83% average improvement.

Table 3.4: Prediction Accuracy Comparison between DOS-ELM and SVR

Machine Learning	**MAPE**			**RMSE**		
	Max	**Min**	**Avg**	**Max**	**Min**	**Avg**
DOS-ELM	0.23	0.10	0.15	29.37	10.92	20.30
SVR	0.34	0.14	0.20	33.75	15.72	23.14
Imp.(%)	31.20	0.50	14.83	30.53	3.32	14.60

3.4.4 Peak Load Reduction

Based on predicted results mentioned above, solar energy can be effectively and optically allocated to smart building. To estimate the proposed solar energy allocation strategy, the convetional static allocation method is used as comparison reference, which immediately allocates all the generated solar energy without storage. As shown in Figure 3.10, the proposed strategy performs better according to the more flat curve which demonstrates the standard deviation of energy consumption profile is lower. Moreover, Table 3.5 shows the reduced load value during peak load period. With different solar PV areas, the proposed solar energy allocation strategy achieves peak load reduction by 6.08% to 18.19%, leading to about 50% cost savings.

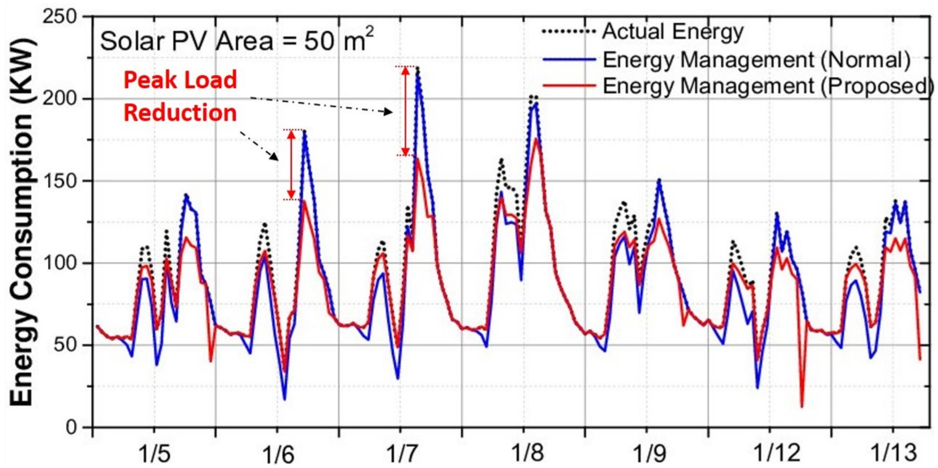

Figure 3.10: Demand response using proposed solar allocation strategy and conventional static allocation method respectively

Table 3.5: Energy Cost Saving and Energy Peak Reduction

	Energy Cost Saving(%)			**Energy Peak Reduction(%)**		
Area	**Max**	**Min**	**Avg**	**Max**	**Min**	**Avg**
25 m^2	83.19	11.59	43.60	16.44	0	6.08
30 m^2	86.26	11.59	48.37	18.30	1.43	9.07
35 m^2	88.46	17.26	51.94	23.31	1.76	15.20
40 m^2	90.10	27.81	51.92	25.32	2.01	16.82
45 m^2	91.38	24.19	49.83	25.32	4.37	17.12
50 m^2	92.41	19.84	45.93	25.32	13.04	18.19

Chapter 4

Lensless Cell Counting System

4.1 Lensless Blood Cell Counting System

The proposed lensless blood cell counting system (LBCS) consists of a poly-dimethylsiloxane (PDMS) microfluidic channel which is bonded on the top surface of a CMOS image sensor (CIS) as shown in Figure 4.1. Through an external syringe pump, cell sample solution can be continuously driven into the microfluidic channel at a typical flow rate of $5\mu L/min$. Meanwhile, a white LED lamp is setup above to illuminate the flowing cells, whose light intensity is designed as $1.5k\ Lux$ at the sensor surface. Then, the cell shadow images will be continuously captured by the CIS underneath and buffered to be processed by machine-learning based SR algorithm. The design of the whole system is introduced as follows.

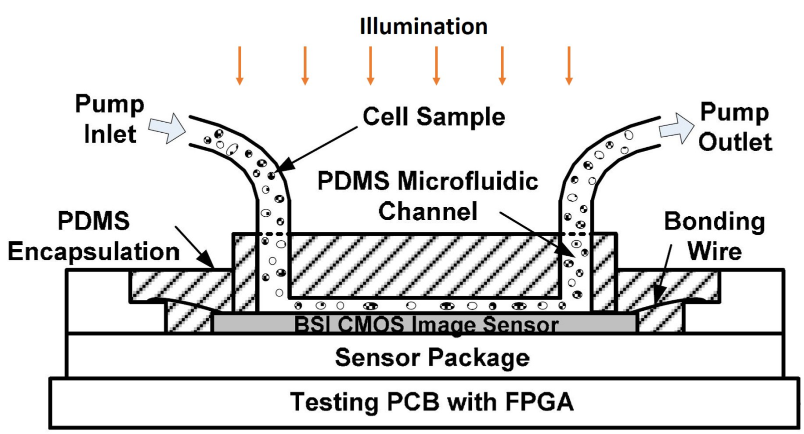

Figure 4.1: General lensless blood cell counting system: cross-sectional view of the lensless system

CMOS Image Sensor There are two kinds of CMOS image sensors employed in the proposed system.

One is a commercial grayscale CMOS image sensor ($MT9M032$, $Onsemi$, San $Jose$, CA, USA) whose pixel size is $2.2\mu m \times 2.2\mu m$ and array size is $1472(H) \times 1096(V)$ with an active pixel area of $3.24mm(H) \times 2.41mm(V)$. Before bonding with the microfluidic channel, the microlens layer is removed by plasma treatment ($PDC-32G$, $Harrick$ $Plasma$, $Ithaca$, NY, USA) and the CIS protection glass is removed by a razor blade. However, as shown in Figure 4.2 a, because a front-side illuminated (FSI) structure is adopted by the pixels of $MT9N032$, a thin PDMS film should be coated on the sensor die. In this way, it can not only protect and encapsulate the sensor top circuit, but also make the sensor surface flatter resulting in tighter bonding. However,

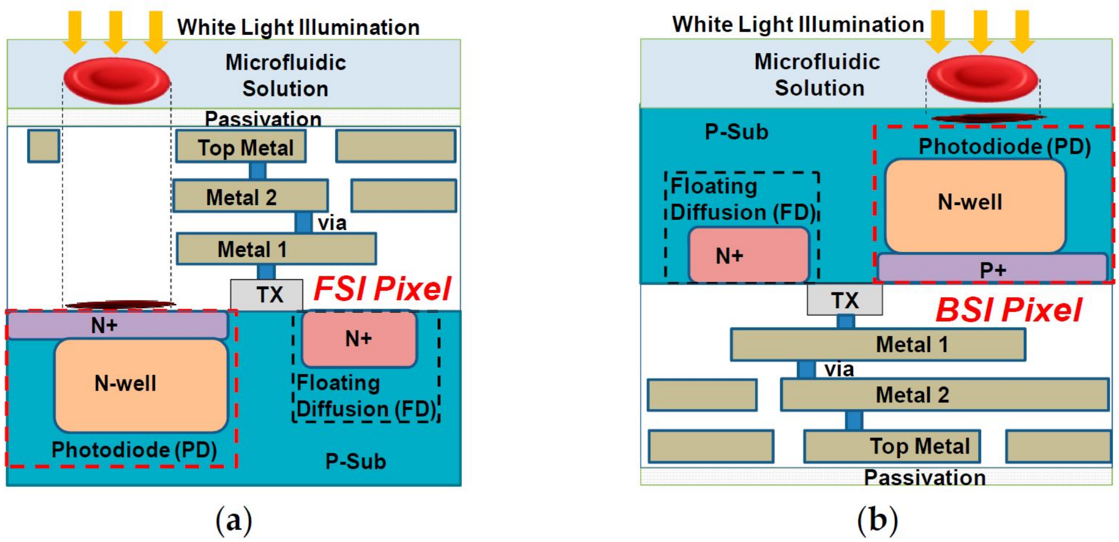

Figure 4.2: Comparison of FSI and BSI CMOS image sensor pixel. (a) FSI pixel: the photodiode is far from cell samples; (b) BSI pixel: the photodiode is closed to the cell samples

the extra PDMS layer causes some problems. First, it will degrade the contrast of lensless shadow image and increase the object distance. Second, because the metal layers which are stackted above photodiodes (PD) suffer light degradation, FSI CIS is low sensitivity.

To overcome the aforementioned problems, a back-side illuminated (BSI) CMOS image sensor is designed with $1.7mm \times 2.2mm$ photosensitive area, $5mm \times 5mm$ die area, a 3.2-Mega pixel array and a $1.1\mu m$ pixel pitch. As shown in Figure 4.2 b, Photodiodes are fabricated on the top layer without metal, which can block incident light. Such a design can make the cell sample directly contact the pixel surface, which achieves quantum efficiency and better sensitivity. Moreover, the top layer of the BSI sensor is a smooth and flat silicon as well as silicon dioxide layer without a microlens. In this way, the microfluidic channel can be integrated directly.

Microfluidic Channel In order to capture flowing cells and avoid missing cells at high rates, it is necessary for the microfluidic channel to fit in the sensor die on a diagonal, which leads to the channel length maximization. The channel length is set as $2.6mm$ for the BSI CIS based prototype and $4.6mm$ for the commercial CIS based prototype. For both prototypes, the channel width is designed as $500\mu m$. When the flowing cell sample is in high concentration, such a relative wide channel will be helpful for alleviating clogging effect. The channel height is designed as $30\mu m$, which is just longer than the diameter of tumor cells and blood cells. Cell samples can thereby flow through in close proximity to the sensor surface, resulting in generating higher contrast lensless images.

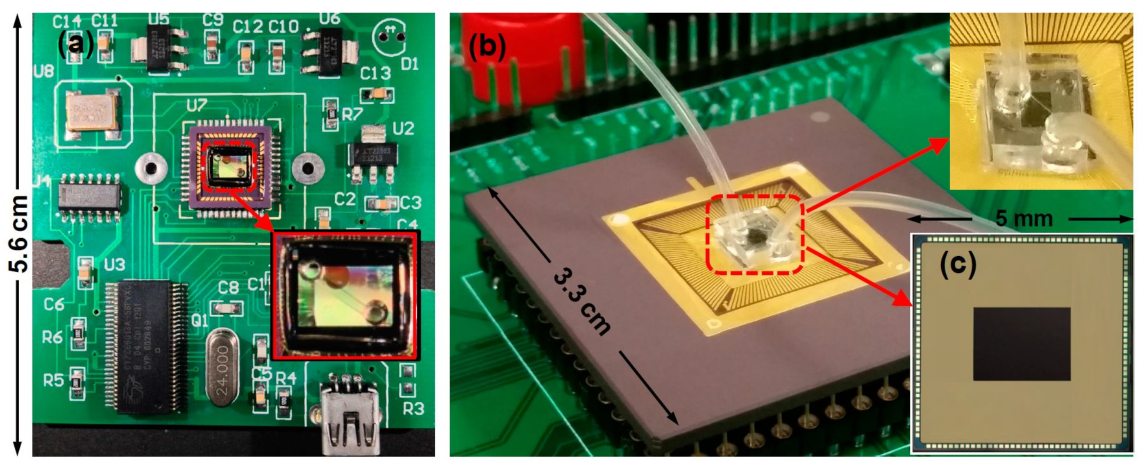

Figure 4.3: Testing board. (a) System with the commercial FSI CIS; (b) Package BSI CIS that is integrated with elastic thin tubing and the microfluidic channel; (c) The custom designed BSI CIS chip

Testing Board As shown in Figure 4.3 a, it's a $5.6cm \times 5.6cm$ printed circuit board (PCB) which is equipped with power supplies for the sensor and one USB interface ($CY7C68013-56\ EZ-USB\ FX2$, $Cypress$, $San\ Jose$, CA, USA) for transferring data. On such a PCB, the commercial CIS chip is soldered. As shown in the inset of Figure 4.3, it's an enlarged figure of a PDMS microfluidic device.

As shown in Figure 4.3 b, the BSI CIS chip is fabricated in a $65nm$ BSI CMOS image sensor process and bonded to a 144-pin ceramic pin grid array (CPGA) package. To build a prototype system, this packaged BSI CIS chip is also mounted on one small field-programmable gate array (FPGA) testing board ($XEM3010$, $Opal\ Kelly$, $Portland$, OR, USA).

4.2 Convolutional Neural Network based Super-Resolution

Convolutional neural network based super-resolution (CNNSR) consists of two steps: off-line training step and on-line testing step. In the off-line training step, the model parameters are optimized, which can correlate the LR images with HR images. In the online testing step, the captured lensless cell image is reconstructed into HR one using the trained model.

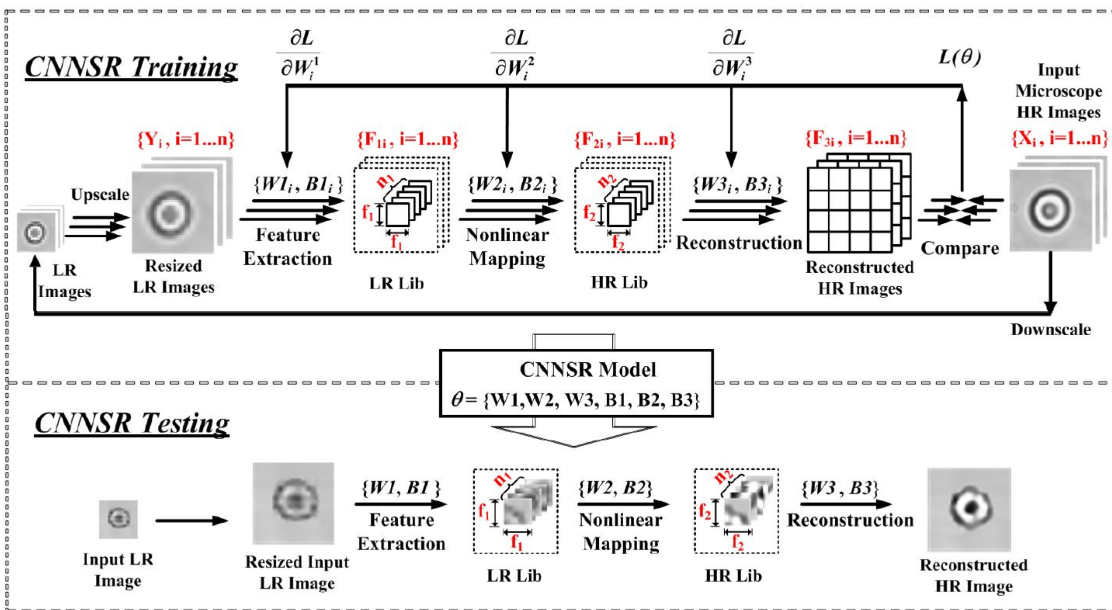

Figure 4.4: Processing flow of CNNSR: off-line training step and on-line testing step

4.2.1 Off-line Training Process

In the off-line training process, consider n training images. Firstly, these LR training images are scaled up to the desired size through bicubic interpolation. These pre-processed images are represented by Y_i. The corresponding ground truth HR images are denoted as X_i. Note the interpolated images Y_i are the same size as X_i. The aim of off-line training step is obtaining an end-to-end function $F(Y_i, \Theta)$ that can improve the LR images Y_i to the HR images $F(Y_i)$ whose resolution is similar as that of X_i. The mean squared error (MSE) acts as the loss function $L(\Theta)$ between Y_i and X_i:

$$L(\Theta) = \frac{1}{n} \sum_{n}^{i=1} \|F(Y_i; \Theta) - X_i\|^2 \tag{4.1}$$

Here, n is denoted as the number of training images. $\Theta = \{W_1, W_2, W_3, B_1, B_2, B_3\}$ represents the group of model parameters which should be estimated by minimizing the loss function.

Training process can be summarized as three steps: feature extraction, non-linear mapping and reconstruction.

First step aims to extract overlapping patches from input LR images Y and represent each one as a high-dimensioned vector. It includes convolutional layer and rectified linear unit (ReLU) function, which can be expressed as $F_i(Y)$:

$$F_1(Y) = max(0, W_1 * Y + B_1) \tag{4.2}$$

here, W_1 can be expressed as $c \times f_1 \times f_1 \times n_1$, which represents n_1 filters of spatial size $f_1 \times f_1$ that convolute the input Y with c channels. B_1 is the bias and can be represented as a n_1-dimensional vector whose each element is associated with the filter. The output of convolutional layer is composed by n_1-dimensional feature maps. The ReLU function applies an elementwise activation function $max(0, x)$ to these filter responses thresholding at zero.

Second step performs non-linear mapping from each n_1-dimensional vector to n_2-dimensional one. It is also comprised by convolutional layer and ReLU function, whose operation is similar as the first step. The operation can expressed as:

$$F_2(Y) = max(0, W_2 * F_1(Y) + B_2) \tag{4.3}$$

here, W_1 represents weights of size $n_1 \times f_2 \times f_2 \times n_2$, which means n_2 filters of spatial size $f_2 \times f_2$ applied in $F_1(Y)$ with n_1 depth. B_2 is a n_2-dimensional bias vector. Conceptually, each output n_2-dimensional vector can represent one HR patch that can be used to reconstruct the final HR image.

Third step can reconstruct HR image by aggregating the above HR patchewise representations. The generated HR image is produced by a convolutional layer and is expected to be as similar as possible to the ground truth HR image X.

$$F_3(Y) = W_3 * F_2(Y) + B_3 \tag{4.4}$$

here, W_3 is weights of size $n_2 \times f_3 \times f_3 \times c$. It represents that n_2 filters of spatial size $f_3 \times f_2$ is used. c is equal to the number of HR image channel. And B_3 represents c-dimensional bias vector. Resulting in producing the recovered full HR image, the filter on a set of feature maps can be considered as averaging the predicted overlapping HR patches.

In order to get the end-to-end mapping function $F(\mathbf{Y}, \Theta)$, network parameters $\Theta = \{W_1, W_2, W_3, B_1, B_2, B_3\}$ should be optimized by minimizing the loss function $L(\Theta)$ through stochastic gradient descent with the standard back-propagation. The weight matrices can be updated:

$$\begin{aligned} \triangle_{i+1} &= 0.9 \times \triangle_i + \eta \times \frac{\partial L}{\partial W_i^l} \\ W_{i+1}^l &= W_i^l + \triangle_{i+1} \end{aligned} \tag{4.5}$$

here i is the index of iterations and l represents the index of layers. $l \in \{1, 2, 3\}$. η is the learning rate.

For the bicubic interpolated image, the magnification factor is implemented as $4\times$. For the first convolutional layer, the number of filters n_1 is 64 and the spatial size of filter $f_1 \times f_i$ is set as 5×5. For the second convolutional layer, the number of filters n_2 is 32 and the filter spatial size $f_2 \times f_2$ is set as 1×1. And the filter size in the third convolutional layer is set as $f_3 = 3$. Moreover, the number of channel c is set as

1 because both the input LR image and ground truth HR image are Grayscale image whose color channel is only 1.

CNNSR Off-line Training

Input: LR cell images Y_i and corresponding HR cell images Y_i
Output: Model parameter Θ=W1, W2, W3, B1, B2, B3
Θ are initialized by drawing randomly from Gaussian Distribution ($\mu = 0$, $\sigma = 0.001$)
For i = 0 to n // n is the number of training image
  For l = 1 to 3 // 3 convolutional layer to tune
  Calculate $F(Y_i)$ based on Equations (13)-(15)
  End For
  Calculate $L(\Theta) = \frac{1}{n}\sum_{n}^{i=1} \|F(Y_i;\Theta) - X_i\|^2$
  If $L(\Theta) < \varepsilon$ ε is closed to zero
  Calculate $\Delta_{i+1} = 0.9 \times \Delta_i + \eta \times \frac{\partial L}{\partial W_i^l}$, $W_{i+1}^l = W_i^l + \Delta_{i+1}$
  End If
End For

CNNSR On-line Testing
Input: LR cell image Y' and network parameter $\Theta = W_1, W_2, W_3, B_1, B_2, B_3$
Output: Corresponding HR cell images FY'
For l = 1 to 3 // 3 convolutional layer network
  Calculate $F(Y')$ based on Equations (13)-(15)
End For

Figure 4.5: Pseudo code for CNNSR

4.2.2 On-line Testing Process

LR cell images are captured by LBCS and upscaled to desired size through bicubic interpolation. When interpolated LR images are input to the CNNSR network with trained grouped parameters Θ, the corresponding HR cell image $F(Y', \Theta)$ can be recovered.

The overall architecture of CNNSR can be described in Figure 4.4. The pseudo code for CNNSR off-line training and on-line testing can be described in Figure 4.5.

4.3 Results and Discussion

In the experiment, both tumor cell and blood cell images are utilized as testing samples for evaluating the performance of CNNSR model. To evaluate the quality of recovered HR images, the structure similarity (SSIM) is selected as a metric. Moreover, another machine-learning based single-frame SR processing method, extreme learning machine based super-resolution (ELMSR) described in detail by Huang al. [19], is also adopted to process the captured cell images in this experiment. Through the comparison of two SR processing machine-learning approaches, the advantages and disadvantages of CNNSR is analyzed and demonstrated.

4.3.1 Sample Preparation

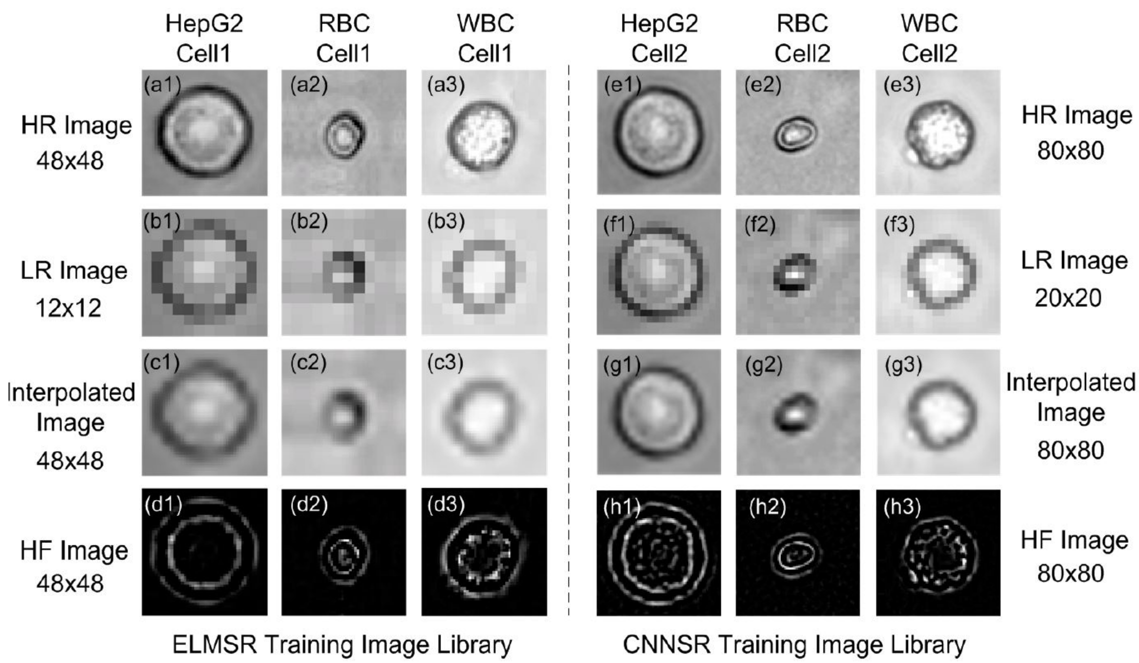

Figure 4.6: Example images in CNNSR and ELMSR training libraries, including of RBC, HepG2, and WBC.

HepG2 tumor cell and red/white blood cell sample solution are prepared for the experiment. Volunteers in Nanyang Technological University (NTU), who signed informed consent forms, donated the blood cell samples with approval of relevant institutional guidelines and laws. The HepG2 tumor cells (*American Type Culture Colletion, Baltimore, MD, USA*) were cultured in Minimum Essential Media (MEM) and grown in a T75 flask in a 5% CO_2 atmosphere and 37 degree Centigrage. The harvested cells were washed and re-suspended in PBS. Moreover, before inputing cell sample solution into the microchannel, all the samples are sonicated by an ultrasonic cleaner ($2510E - DTH$, *Branson Ultrasonics, Danbury, CT, USA*) for 10 minutes, in order to prevent cell aggregation.

4.3.2 Off-line Super-resolution Training

In this section, the building off-line training image libraries will be introduced. First, a microscope camera ($OlympusIX71, Tokyo, Japan$) is used to capture HR training images of blood cells and HepG2 tumor cells. As shown in Figure 4.6 $a1-a3, e1-e3$, these HR images are collected in HR image library. Because of two prototypes with different CIS, these original HR images should be saved as two sizes, 80×80 and 48×48 corresponding respectively to the CNNSR training image library and ELMSR training image library.

Next, according to the enhancement factor of four, the 48×48 HR images are down-sampled to 12×12 LR images in Figure 4.6 $b1-b3$, and the 80×80 HR images are down-sampled to20×20 LR images in Figure 4.6 $f1-f3$. Then, these down-sampled LR images are resized to 48×48 and 80×80 through bicubic interpolation, as shown in Figure 4.6 c1-c3, g1-g3. Since bicubic interpolation operation can not recover high-frequency (HF) components, detailed structure information in those interpolated images has been lost already. Then, by subtracting the original HR cell images with corresponding interpolated one, the HF components of the training cell image can be obtained. In the HF components images in Figure 4.6 d1-d3, h1-h3, specific features in each cell types are obviously different. Through above four steps, the training sample libraries for both CNNSR and ELMSR can be generated. To build the library, thirty HR cell images for each cell type are selected.

Finally, based on the built training sample libraries, model can be trained using CNNSR and ELMSR approaches respectively.

4.3.3 On-line Super-resolution Testing

In this section, the on-line testing results of both CNNSR and ELMSR will be shown and analyzed. Based on built library and trained model, the testing process can be performed with captured lensless LR images inputting. Since there are two CMOS image sensors of pixel sizes ($2.2\mu m$ vs. $1.1\mu m$) in the lensless imaging systems, the directly captured LR images can be compared as shown in Figure 4.7 a, d. It is obvious that the captured LR RBC in Figure 4.7 $d1$ is much clearer than that of Figure 4.7 $a2$, because the pixel pitch of BSI CIS is smaller than the commercial FIS CIS. Moreover, the diameter of LR RBC images is 4 pixels using FSI CIS while the LR RBC images cover 8 pixel using BSI CIS. According to these results, CIS of the smaller pixel

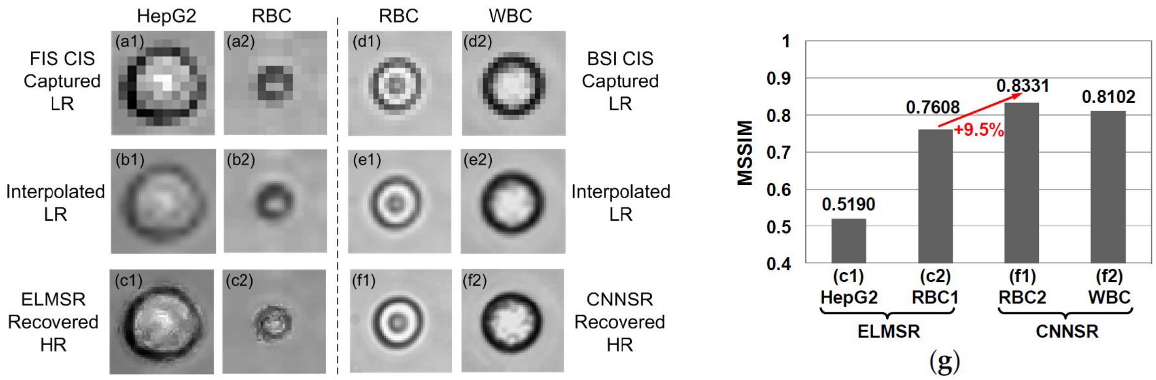

Figure 4.7: Example of HepG, RBC and WBC images in CNNSR and ELMSR testing.

pitch can be validated to perform better for generating LR images of higher spatial resolution. Then, the interpolated LR images will be generated as shown in Figure 4.7 b, e. Next, these interpolated LR images are input into the trained model. As shown in Figure 4.7 $c1, c2$, the HR images of the RBC cell and HepG2 cell can be recovered through ELMSR model. And the HR images of the WBC cell and RBC cell can be reconstructed by CNNSR model as shown in Figure 4.7 $f1, f2$.

Through the comparison between interpolated LR images in Figure 4.7 b, e and recovered HR images in Figure 4.7 c, f, it is obvious that more cell internal and edge information can be displayed in recovered HR images no matter which SR approaches are used. Moreover, comparing the performance of ELMSR, the CNNSR can perform better in HR image reconstruction with less noise. As shown in Figure 4.7 $c2$, there is some blur effect in the cell edge recovered by ELMSR, while the cell membrane and nucleolus in the recovered cell image by CNNSR can be seen clearly as shown in Figure 4.7 f1,f2.

In Figure 4.7 g, the performance of CNNSR and ELMSR is evaluated by comparing the MSSIM metric. The MSSIM of HepG2 in Figure 4.7 $c1$, RBC in Figure 4.7 $c2, f1$ and WBC in Figure 4.7 $f2$ are 0.5190, 0.7608, 0.8331 and 0.8102 respectively. As shown in Figure 4.7 g, CNNSR can achieve 9.5% improvement over ELMSR on SR improvement quality. This advantage of CNNSR is probably resulted from the filter optimization in all the layers of CNN network, while there is no such joint optimization for ELMSR model training. Note that even though the input LR images for CNNSR and ELMSR approaches are obtained from different CMOS image sensors, there is no effect on results evaluation because the recovered images are only compared with their original HR images in corresponding off-line training sample libraries.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In conclusion, this thesis has investigated machine learning based data analytics for energy management in smart buildings and lensless cell counting system respectively. From machine learning algorithm perspective, this thesis has proposed online sequential extreme learning machine with distribution architecture, which can update model continuously. Then, convolutional neural network has been employed to perform image resolution improvement. From application perspective, this thesis has designed and built IoT system for smart home/building field and biomedical field: smart building energy management system and microfluidic lensless blood-cell counting system.

For energy management in smart residential buildings, we have firstly performed short-term load forecasting in a distributed real-time pattern. To perform accurate load forecasting, we have considered occupants behavior, indoor and outdoor environments for model building and updating. Furthermore, based on such load forecasting, we have performed solar energy allocation to reduce peak power consumption and save bills for users. Experimental results have shown that the distributed machine learning algorithm (DOS-ELM) is 14.83% more accurate for load forecasting than the traditional machine learning (SVR) method. By applying our proposed dynamic solar energy allocation algorithm, 51.94% more energy saving and 15.20% more peak load reduction can be achieved comparing to the static energy allocation method.

For microfluidic lensless blood-cell counting system, captured cell images are low-resolution without detailed information, resulting in a challenge for cell counting and reorganization. To solve this problem, convolutional neural network based super-resolution (CNNSR) is employed to recover HR images with low cost of computation resource and without limitation of system throughput. Based on the built system with commercial CMOS image sensors and custom designed backside-illuminate CMOS sensors, high-resolution cell images are reconstructed through CNNSR with comparison of ELMSR. The experiment results show that the cell resolution is improved by $4\times$, and CNNSR has 9.5% improvement over the ELMSR on resolution enhancing performance. The CNNSR can be matched well with the developed lensless blood cell counting system.

5.2 Recommendation for Future Work

5.2.1 IoT for Residential Building

Since the energy demand prediction is the base of energy management, it is important to further improve the accuracy of short-term load forecasting in future work. Accuracy of energy profile extraction can be improved according to several work plans as follows:

- The current model is updated day by day through online sequential machine learning. To obtain an energy profile with higher resolution, the solution can update the model hour by hour. In this way, the predicted load value can be adjusted with the dynamic changes of energy consumption.

- In current work, the energy profile is independent to occupant behavior profile which is represented by positioning information. Actually, the occupants behavior has an important effect on energy consumption. It is difficult to work out their quantitative relationship. A further task aims to analyze the occupant behavior and extract behavior pattern as a feature of load forecasting.

- For current IPS developed based on Wi-Fi communication technique, the occupants' positioning can be only detected when Wi-Fi is turned on as well as stable. It is a problem how to detect occupants' position when the Wi-Fi is turned off. To solve this problem, auxiliary communication technology can be adopted for IoT devices or more functions of mobile phone can be explored. For example, when the Wi-Fi of smart devices is turned off, GSM can be a solution because GSM signal collector can be integrated with existing antenna to monitor phone signal.

5.2.2 IoT for Bio-medicine

Rapid diagnosis is expected for on-site disease monitoring, leading to that the bio-medical IoT system should output data analytics result in real time. For current work, pre-processing operation and non-linear mapping are regarded as the most time-consuming part. Such a convolutional neural network is strongly needed to accelerate. So future work is expected to focus on further investigating the convolutional neural network for faster image super-resolution.

Appendix A

Publication List

A.1 Journal paper

:

1. Hantao Huang, Yuehua Cai, **Hang Xu**, Suleman Khalid Rai, Zichuan Liu and Hao Yu*, “A Multi-agent Minority-game based Demand-response Management of Smart Buildings towards Peak Load Reduction”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (TCAD) 2016.

2. Xiwei Huang, Yu Jiang, Xu Liu, **Hang Xu**, Zhi Han, Hailong Rong, Haiping Yang, Mei Yan*, Hao Yu* ” Machine Learning Based Single-Frame Super-Resolution Processing for Lensless Blood Cell Counting”, *Sensors* 2016 Nov 2; 16(11):1836

3. Xu Liu, *Memeber, IEEE*, Xiwei Huang, *Memeber, IEEE*, Yu Jiang, **Hang Xu**, Jing Guo, Han Wei Hou, Mei Yan, Hao Yu, *Senior Memeber, IEEE*, ”A Microfluidic Cytometer for Complete Blood Count with a 3.2-Megapixel, 1.1-m-pitch Super-Resolution Image Sensor in 65-nm BSI CMOS”, *IEEE Transactions on Biomedical Circuits and Systems* 2016

A.2 Conference paper

:

1. **Hang Xu**, Hantao Huang, Rai Suleman Khalid and Hao Yu*, “Distributed Machine Learning based Smart-grid Energy Management with Behavior Cognition”, *IEEE International Conference on Smart Grid Communications* (SmartGridComm) 2016.

2. Xiwei Huang, Jiang Yu, **Hang Xu**, Xu Liu, Han Wei Hou, Mei Yan and Hao Yu*, "A Convolutional Neural Network based Single-frame Super-Resolution for Lenless Blood Cell Counting", *IEEE BioMedical Circuits and Systems Conference* (BioCAS) 2016.

3. Zichuan Liu, Yi Feihu, **Hang Xu**, Lamees Nasser, Philippe Coquet, Thomas Boudier and Hao Yu*, "NucleiNet: A Convolutional Encoder-decoder Network for Bio-image Denoising", *IIEEE Engineering in Medicine & Biology Society* (EMBC) 2017.

Bibliography

[1] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[2] N. Lu, T. Taylor, W. Jiang, J. Correia, L. R. Leung, and P. C. Wong, "The temperature sensitivity of the residential load and commercial building load," in *IEEE Power & Energy Society General Meeting*. IEEE, 2009.

[3] P. Areekul, T. Senjyu, H. Toyama, and A. Yona, "Notice of violation of ieee publication principles a hybrid arima and neural network model for short-term price forecasting in deregulated market," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 524–530, 2010.

[4] K. Nose-Filho, A. D. P. Lotufo, and C. R. Minussi, "Short-term multinodal load forecasting using a modified general regression neural network," *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2862–2869, 2011.

[5] Y.-M. Wi, S.-K. Joo, and K.-B. Song, "Holiday load forecasting using fuzzy polynomial regression with weather feature selection and adjustment," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 596–603, 2012.

[6] J. W. Taylor and R. Buizza, "Neural network load forecasting with weather ensemble predictions," *IEEE Transactions on Power systems*, vol. 17, no. 3, pp. 626–632, 2002.

[7] T. Senjyu, H. Takara, K. Uezato, and T. Funabashi, "One-hour-ahead load forecasting using neural network," *IEEE Transactions on power systems*, vol. 17, no. 1, pp. 113–118, 2002.

[8] N. Kandil, R. Wamkeue, M. Saad, and S. Georges, "An efficient approach for short term load forecasting using artificial neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 8, pp. 525–530, 2006.

[9] M. Ghiassi, D. K. Zimbra, and H. Saidane, "Medium term system load forecasting with a dynamic artificial neural network model," *Electric Power Systems Research*, vol. 76, no. 5, pp. 302–316, 2006.

[10] O. A. Carpinteiro, R. C. Leme, A. C. Z. de Souza, C. A. Pinheiro, and E. M. Moreira, "Long-term load forecasting via a hierarchical neural model with time integrators," *Electric Power Systems Research*, vol. 77, no. 3, pp. 371–378, 2007.

[11] N. Ye, Y. Liu, and Y. Wang, "Short-term power load forecasting based on svm," in *World Automation Congress (WAC)*. IEEE, 2012.

[12] R. Zhang, Z. Y. Dong, Y. Xu, K. Meng, and K. P. Wong, "Short-term load forecasting of australian national electricity market by an ensemble model of extreme learning machine," *IET Generation, Transmission & Distribution*, vol. 7, no. 4, pp. 391–397, 2013.

[13] W. Jung, J. Han, J.-W. Choi, and C. H. Ahn, "Point-of-care testing (poct) diagnostic systems using microfluidic lab-on-a-chip technologies," *Microelectronic Engineering*, vol. 132, pp. 46–57, 2015.

[14] C. van Berkel, J. D. Gwyer, S. Deane, N. Green, J. Holloway, V. Hollis, and H. Morgan, "Integrated systems for rapid point of care (poc) blood cell analysis," *Lab on a Chip*, vol. 11, no. 7, pp. 1249–1255, 2011.

[15] X. Mao, A. A. Nawaz, S.-C. S. Lin, M. I. Lapsley, Y. Zhao, J. P. McCoy, W. S. El-Deiry, and T. J. Huang, "An integrated, multiparametric flow cytometry chip using microfluidic drifting based three-dimensional hydrodynamic focusing," *Biomicrofluidics*, vol. 6, no. 2, p. 024113, 2012.

[16] S.-C. Lin, P.-W. Yen, C.-C. Peng, and Y.-C. Tung, "Single channel layer, single sheath-flow inlet microfluidic flow cytometer with three-dimensional hydrodynamic focusing," *Lab on a Chip*, vol. 12, no. 17, pp. 3135–3141, 2012.

[17] G. Zheng, S. A. Lee, S. Yang, and C. Yang, "Sub-pixel resolving optofluidic microscope for on-chip cell imaging," *Lab on a Chip*, vol. 10, no. 22, pp. 3125–3129, 2010.

[18] S. A. Lee, R. Leitao, G. Zheng, S. Yang, A. Rodriguez, and C. Yang, "Color capable sub-pixel resolving optofluidic microscope and its application to blood cell imaging for malaria diagnosis," *PloS one*, vol. 6, no. 10, p. e26127, 2011.

[19] X. Huang, J. Guo, X. Wang, M. Yan, Y. Kang, and H. Yu, "A contact-imaging based microfluidic cytometer with machine-learning for single-frame super-resolution processing," *PloS one*, vol. 9, no. 8, p. e104539, 2014.

[20] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE signal processing magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[21] X. Huang, H. Yu, X. Liu, Y. Jiang, and M. Yan, "A single-frame superresolution algorithm for lab-on-a-chip lensless microfluidic imaging," *IEEE Design & Test*, vol. 32, no. 6, pp. 32–40, 2015.

[22] T. Wang, X. Huang, Q. Jia, M. Yan, H. Yu, and K. S. Yeo, "A super-resolution cmos image sensor for bio-microfluidic imaging," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2012.

[23] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *International journal of computer vision*, vol. 40, no. 1, pp. 25–47, 2000.

[24] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 2, p. 12, 2011.

[25] J. Yang, Z. Lin, and S. Cohen, "Fast image super-resolution based on in-place example regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[26] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[27] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[28] K. Ashton, "That internet of things thing," *RFiD Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[29] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[30] L. Da Xu, "Enterprise systems: state-of-the-art and future trends," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 630–640, 2011.

[31] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the internet of things," *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 147–159, 2011.

[32] J. Hyuk Park, S. Gritzalis, C.-H. Hsu, R. Roman, and J. Lopez, "Integrating wireless sensor networks and the internet: a security analysis," *Internet Research*, vol. 19, no. 2, pp. 246–259, 2009.

[33] C. Alcaraz and J. Lopez, "A security analysis for wireless sensor mesh networks in highly critical systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 4, pp. 419–428, 2010.

[34] A. J. Jara, M. A. Zamora-Izquierdo, and A. F. Skarmeta, "Interconnection framework for mhealth and remote monitoring based on the internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 47–65, 2013.

[35] E. A. Feinberg and D. Genethliou, "Load forecasting," in *Applied mathematics for restructured electric power systems*. Springer, 2005.

[36] C. Xia, J. Wang, and K. McMenemy, "Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 7, pp. 743–750, 2010.

[37] S.-H. Ling, F. H.-F. Leung, H. Lam, and P. K.-S. Tam, "Short-term electric load forecasting based on a neural fuzzy network," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 6, pp. 1305–1316, 2003.

[38] S. C. Pandian, K. Duraiswamy, C. C. A. Rajan, and N. Kanagaraj, "Fuzzy approach for short term load forecasting," *Electric Power Systems Research*, vol. 76, no. 6, pp. 541–548, 2006.

[39] A. Al-Kandari, S. Soliman, and M. El-Hawary, "Fuzzy short-term electric load forecasting," *International Journal of Electrical Power & Energy Systems*, vol. 26, no. 2, pp. 111–122, 2004.

[40] A. D. Papalexopoulos and T. C. Hesterberg, "A regression-based approach to short-term system load forecasting," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1535–1547, 1990.

[41] T. Haida and S. Muto, "Regression based peak load forecasting using a transformation technique," *IEEE Transactions on Power Systems*, vol. 9, no. 4, pp. 1788–1794, 1994.

[42] W. Charytoniuk, M. Chen, and P. Van Olinda, "Nonparametric regression based short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 725–730, 1998.

[43] M. Cho, J. Hwang, and C. Chen, "Customer short term load forecasting by using arima transfer function model," in *Energy Management and Power Delivery (EMPD)*. IEEE, 1995.

[44] J. Fan and J. McDonald, "A real-time implementation of short-term load forecasting for distribution power systems," *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 988–994, 1994.

[45] S. Kiartzis, A. Bakirtzis, J. Theocharis, and G. Tsagas, "A fuzzy expert system for peak load forecasting. application to the greek power system," in *Electrotechnical Conference*. IEEE, 2000.

[46] D. Ranaweera, N. Hubele, and G. Karady, "Fuzzy logic for short term load forecasting," *International journal of electrical power & energy systems*, vol. 18, no. 4, pp. 215–222, 1996.

[47] A. Bakirtzis, V. Petridis, S. Kiartzis, M. Alexiadis, and A. Maissis, "A neural network short term load forecasting model for the greek power system," *IEEE Transactions on power systems*, vol. 11, no. 2, pp. 858–863, 1996.

[48] T. Chow and C.-T. Leung, "Nonlinear autoregressive integrated neural network model for short-term load forecasting," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 143, no. 5, pp. 500–506, 1996.

[49] H. Mori and N. Kosemura, "Optimal regression tree based rule discovery for short-term load forecasting," in *Power Engineering Society Winter Meeting*. IEEE, 2001.

[50] B. Dong, C. Cao, and S. E. Lee, "Applying support vector machines to predict building energy consumption in tropical region," *Energy and Buildings*, vol. 37, no. 5, pp. 545–553, 2005.

[51] Q. Li, Q. Meng, J. Cai, H. Yoshino, and A. Mochida, "Applying support vector machine to predict hourly cooling load in the building," *Applied Energy*, vol. 86, no. 10, pp. 2249–2256, 2009.

[52] Z. Hou and Z. Lian, "An application of support vector machines in cooling load prediction," in *Intelligent Systems and Applications (ISA)*. IEEE, 2009.

[53] Q. Li, P. Ren, and Q. Meng, "Prediction model of annual energy consumption of residential buildings," in *Advances in Energy Engineering (ICAEE)*. IEEE, 2010.

[54] S. Cheng, J. Yan, D. Zhao, Q. Wang, and H. Wang, "Short-term load forecasting method based on ensemble improved extreme learning machine [j]," *Journal of Xi'an Jiaotong University*, vol. 2, p. 029, 2009.

[55] M. Severini, S. Squartini, M. Fagiani, and F. Piazza, "Energy management with the support of dynamic pricing strategies in real micro-grid scenarios," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015.

[56] T. Hong, "Occupant behavior: impact on energy use of private offices," in *ASim 2012-1st Asia conference of International Building Performance Simulation Association., Shanghai, China, 11/25/12-11/27/12*, 2014.

[57] O. G. Santin, L. Itard, and H. Visscher, "The effect of occupancy and building characteristics on energy use for space and water heating in dutch residential stock," *Energy and buildings*, vol. 41, no. 11, pp. 1223–1232, 2009.

[58] C. M. Clevenger and J. Haymaker, "The impact of the building occupant on energy modeling simulations," in *Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada*. Citeseer, 2006.

[59] T. Hong, S. C. Taylor-Lange, S. DOca, D. Yan, and S. P. Corgnati, "Advances in research and applications of energy-related occupant behavior in buildings," *Energy and Buildings*, vol. 116, pp. 694–702, 2016.

[60] P. Warren and L. Parkins, "Window-opening behaviour in office buildings," *Building Services Engineering Research and Technology*, vol. 5, no. 3, pp. 89–101, 1984.

[61] J. L. Hensen and R. Lamberts, *Building performance simulation for design and operation*. Routledge, 2012.

[62] Y. Zhang and P. Barrett, "Factors influencing occupants blind-control behaviour in a naturally ventilated office building," *Building and Environment*, vol. 54, pp. 137–147, 2012.

[63] M. Foster and T. Oreszczyn, "Occupant control of passive systems: the use of venetian blinds," *Building and Environment*, vol. 36, no. 2, pp. 149–155, 2001.

[64] F. Haldi and D. Robinson, "Adaptive actions on shading devices in response to local visual stimuli," *Journal of Building Performance Simulation*, vol. 3, no. 2, pp. 135–153, 2010.

[65] A. Mahdavi, A. Mohammadi, E. Kabir, and L. Lambeva, "Occupants' operation of lighting and shading systems in office buildings," *Journal of Building Performance Simulation*, vol. 1, no. 1, pp. 57–65, 2008.

[66] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[67] A. Vilamovska, E. Hatziandreu, H. Schindler, C. van Oranje-Nassau, H. de Vries, and J. Krapels, "Study on the requirements and options for rfid application in healthcare identifying areas for radio frequency identification deployment in healthcare delivery: a review of relevant literature," *A Review of Relevant Literature, Rand Corporation*, 2009.

[68] S. Peled and Y. Yeshurun, "Superresolution in mri: application to human white matter fiber tract visualization by diffusion tensor imaging," *Magnetic resonance in medicine*, vol. 45, no. 1, pp. 29–35, 2001.

[69] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. Bhatia, A. M. S. M. de Marvao, T. Dawes, D. ORegan, and D. Rueckert, "Cardiac image super-resolution with global correspondence using multi-atlas patchmatch," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2013.

[70] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes, and R. M. Mersereau, "Eigenface-domain super-resolution for face recognition," *IEEE transactions on image processing*, vol. 12, no. 5, pp. 597–606, 2003.

[71] M. Thornton, P. M. Atkinson, and D. Holland, "Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping," *International Journal of Remote Sensing*, vol. 27, no. 3, pp. 473–491, 2006.

[72] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[73] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: a benchmark," in *European Conference on Computer Vision.* Springer, 2014.

[74] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *Computer Vision and Pattern Recognition (CVPR).* IEEE, 2008.

[75] R. Fattal, "Image upsampling via imposed edge statistics," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 95.

[76] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[77] H. He and W.-C. Siu, "Single image super-resolution using gaussian process regression," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011.

[78] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, 2010.

[79] L. Sun and J. Hays, "Super-resolution from internet-scale scene matching," in *Computational Photography (ICCP)*. IEEE, 2012.

[80] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1596–1610, 2007.

[81] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.

[82] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[83] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*. Citeseer, 1990.

[84] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[85] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[86] Uflfl tutorial: Convolutional neural network. [Online]. Available: http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/

[87] Cs231n convolutional neural networks for visual recognition. [Online]. Available: http://cs231n.github.io/convolutional-networks/

[88] Deeplearning 0.1 documentation: Convolutional neural networks (lenet). [Online]. Available: http://deeplearning.net/tutorial/lenet.html

[89] H. Huang, Y. Cai, H. Xu, and H. Yu, "A multi-agent minority-game based demand-response management of smart buildings towards peak load reduction," *IEEE Transactions on Computer-aided Design*, 2015.

[90] S. Liao, A. K. Jain, and S. Z. Li, "Partial face recognition: Alignment-free approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1193–1205, 2013.

[91] C. Zhang, W. Wu, H. Huang, and H. Yu, "Fair energy resource allocation by minority game algorithm for smart buildings," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE).* IEEE, 2012.

[92] J. Wang, K. Li, Q. Lv, H. Zhou, and L. Shang, "Hybrid energy storage system integration for vehicles," in *16th ACM/IEEE international symposium on Low power electronics and design.* ACM, 2010.

[93] A. Kailas, V. Cecchi, and A. Mukherjee, "A survey of communications and networking technologies for energy management in buildings and home automation," *Journal of Computer Networks and Communications*, vol. 2012, 2012.

[94] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural networks*, vol. 17, no. 6, pp. 1411–1423, 2006.

[95] G.-B. Huang, N.-Y. Liang, H.-J. Rong, P. Saratchandran, and N. Sundararajan, "On-line sequential extreme learning machine." *Computational Intelligence*, vol. 2005, pp. 232–237, 2005.

[96] G. H. Golub and C. F. Van Loan, *Matrix computations.* JHU Press, 2012, vol. 3.

[97] H. Huang, Y. Cai, and H. Yu, "Distributed-neuron-network based machine learning on smart-gateway network towards real-time indoor data analytics," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE).* IEEE, 2016.

[98] I. Richardson, M. Thomson, and D. Infield, "A high-resolution domestic building occupancy model for energy demand simulations," *Energy and buildings*, vol. 40, no. 8, pp. 1560–1566, 2008.

[99] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, “Smart*: An open data set and tools for enabling research in sustainable homes,” *SustKDD, August*, vol. 111, p. 112, 2012.

[100] Pricing strategy for new hampshire electric co-op. [Online]. Available: http://www.oca.nh.gov/Advisory%20Board/ArchivedMinutes/20150727Mtg/NHEC%20Presentation%20re%20AMI%20DP%20Prepaid%207-27-15.pdf