Development of an IoT Driven Building Environment for Prediction of Electric Energy Consumption

Guneet Bedi, *Member, IEEE*, Ganesh Kumar Venayagamoorthy, *Senior Member, IEEE*, and Rajendra Singh, *Fellow, IEEE*

***Abstract*—This paper demonstrates the importance integrating Internet of Things (IoT) devices and technologies for efficient energy management in a building environment. An Elman recurrent neural network (RNN) model and an exponential model are developed for electric energy consumption prediction in an IoT driven building environment. The models predict electric energy consumption by electric loads in the near-future by (1) recognizing the existence of a relationship between the net electric energy consumption of the building's electric loads and the ambient temperature along with the occupancy state of the building; and (2) employing the detected relationship to predict electric energy consumption using the forecasted temperature and scheduled occupancy state. The building environment under consideration is the Real-Time Power and Intelligent Systems (RTPIS) laboratory integrated with intelligent monitoring and control capabilities using IoT devices and technologies. The electric loads under consideration include heating, ventilation, and air conditioning (HVAC) units and light panels. The developed Elman RNN and exponential models are also compared. These prediction capabilities are beneficial in overcoming variabilities in electric energy consumption by supplying electric energy as needed to meet the demands of the electric loads, thereby minimizing wasted electric energy, reducing carbon emissions, and generating cost savings.**

***Index Terms*—Computational intelligence; electric energy consumption prediction; Elman recurrent neural network; Internet of Things; and smart building**

This paper was originally submitted for review on April 28, 2019. This work is supported in part by the US National Science Foundation (NSF) under grant #1312260 and the Duke Energy Distinguished Professor Endowment Fund. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF and Duke Energy Foundation.

Guneet Bedi was formerly with the Real-Time Power and Intelligent Systems Laboratory, Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, 29634 USA (email: gbedi@clemson.edu).

Ganesh Kumar Venayagamoorthy is with the Real-Time Power and Intelligent Systems Laboratory, Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA, and also with the School of Engineering, University of KwaZulu–Natal, Durban 4041, South Africa (e-mail: gkumar@ieee.org).

Rajendra Singh is with the Real-Time Power and Intelligent Systems Laboratory, Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, 29634 USA (email: srajend@clemson.edu).

I. INTRODUCTION

CURRENT energy challenges including inefficient energy management, wasted energy resources, and expensive energy costs need to be addressed to meet the growing demand for clean, affordable, and sustainable energy [1-4]. Energy generation is currently controlled based on energy demands of electric loads in near real-time. In this case, maintaining reserve energy resources operational at all times for generating and supplying excess energy to account for sporadic increases in energy demands is an inefficient and unsustainable approach. Electric energy consumption prediction capabilities are needed to reduce the amount of energy wasted from maintaining reserve resources operational at all times. With prediction capabilities, the electric energy consumption can be estimated for the near future and thus, the required amount of energy can be generated and supplied as needed to meet the demands of the electric loads, which minimizes wasted energy, reduces carbon emissions, and generates cost savings [5].

Two prediction models viz. Elman recurrent neural network (RNN) model and exponential model have been developed in this paper. Clemson University's Real-Time Power and Intelligent Systems (RTPIS) laboratory [6], with the integration of IoT devices and technologies, serves as the building case study environment. The developed prediction models predict the electric energy consumption of the RTPIS laboratory loads (heating, ventilation, and air conditioning (HVAC) units and light panels) in real-time and near future.

The objectives of this paper are as follows:

- Development of an IoT driven building case study environment
- Development and application of Elman RNN and exponential prediction models in the IoT driven building environment for electric energy consumption prediction

The rest of the paper is organized as follows: Section II describes the development of the IoT driven building case study environment and the measurement data obtained from IoT devices. Section III discusses the electric energy consumption prediction problem and describes the development of Elman RNN and exponential prediction models. The electric energy prediction results obtained from the prediction models, their

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2975847, IEEE Internet of Things Journal

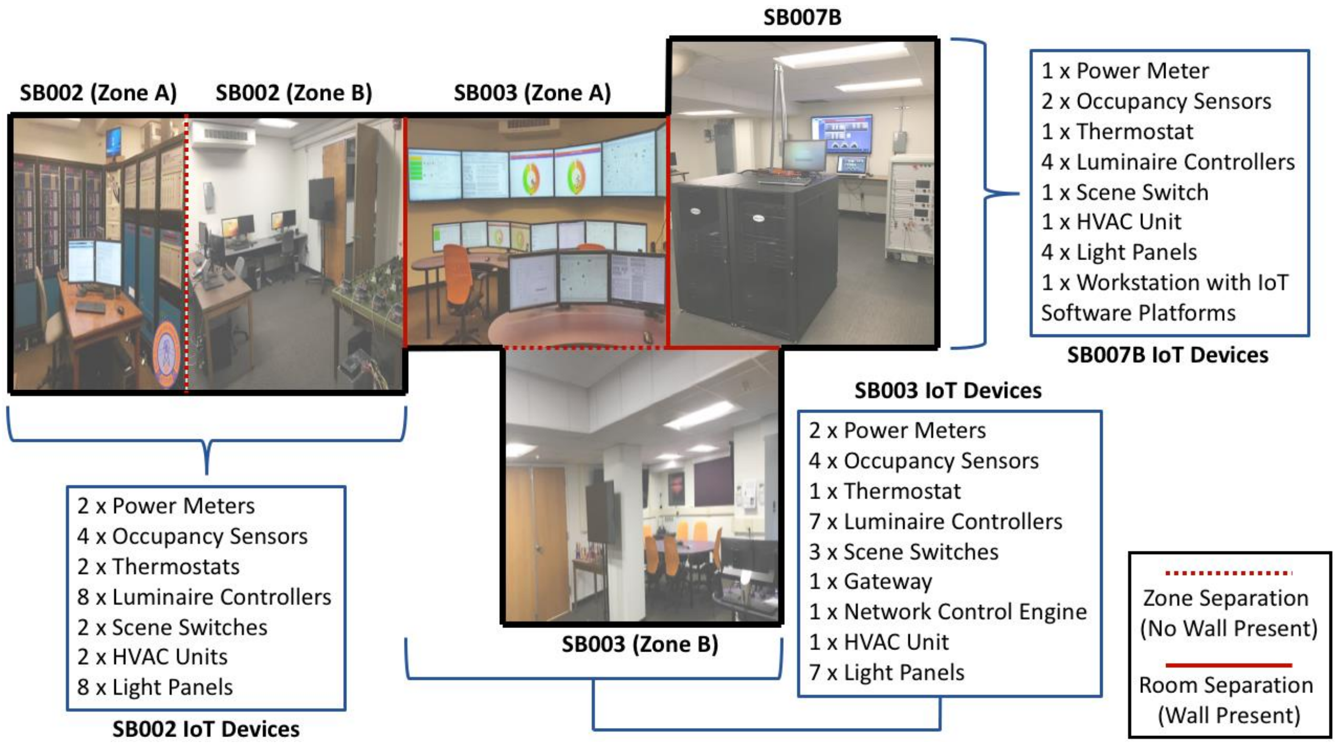

Fig. 1. RTPIS laboratory layout

comparison, discussions, and application are presented in Section IV. Finally, the conclusion is given in Section V.

II. IoT Driven Building Case Study Environment

A. Real-Time Power and Intelligent Systems Laboratory

The building environment under consideration for the deployment of IoT devices and technologies is the RTPIS Laboratory. RTPIS laboratory is a premier world class research, education and innovation-ecosystem laboratory for smart grid technologies. It is housed in the sub-basement of Riggs Hall at Clemson University, Clemson, South Carolina, USA. The RTPIS laboratory comprises of the following three specialized laboratories each housed in a separate room [6]:

- SB002: Real-Time Grid Simulation Laboratory
- SB003: Situational Intelligence Laboratory
- SB007B: Digital Laboratory

The layout of the RTPIS laboratory is shown in Fig. 1.

B. Deployment of IoT Devices and Technologies in RTPIS Laboratory

1) IoT Devices and Technologies

The IoT devices and technologies deployed in the RTPIS laboratory include smart power meters, occupancy sensors, smart thermostats, smart luminaire controllers, smart switches, a gateway, and a network control engine (NCE) (Fig. 2) for monitoring and optimized control of the electric loads viz. HVAC units and light panels (Fig. 2). Each HVAC unit and light panel has a dedicated thermostat and luminaire controller, respectively. A brief description of each installed IoT device is provided below.

Energy consumption data from the electric loads in the RTPIS laboratory was measured using the Setra Power Patrol power meters [7]. These are three-phase power meters that work with Rogowski Coils and communicate either through Ethernet (Building Automation and Control Network (BACnet) IP/ Modbus TCP) or through RS-485 serial connection (BACnet MS/TP / Modbus). The Power Patrol BACnet/Modbus Power Meters offer the following benefits:

- Small form factor that makes it easy to mount inside or outside the panel
- Rogowski and CT compatible, providing added flexibility
- Easy to configure through computer's USB port
- Supports both BACnet and Modbus-based communication
- No external power required since power meter is line powered from 80-600V

Johnson Controls thermostats [8] controlled the switching off and on of the HVAC units based on user-specified temperature setpoint. These thermostats offer the following benefits:

- Remote monitoring and temperature setpoint management
- Remote wireless occupancy scheduling
- Programmable temperature and control schedule
- BACnet compatible
- Maintenance-free operation
- Reliable zone comfort
- Enhanced energy economy
- Maximized energy savings without sacrificing user comfort

Each light panel in the RTPIS laboratory had an Audacy luminaire controller [9] installed in it, which controlled the switching on and off as well as 0-10V dimming of lights. The luminaire controllers are AC-powered, BACnet compatible, operate in highly reliable 915 MHz spectrum, and easy to install.

Audacy scene switches [10] were wall-mounted in each room of the RTPIS laboratory. These switches were pre-configured

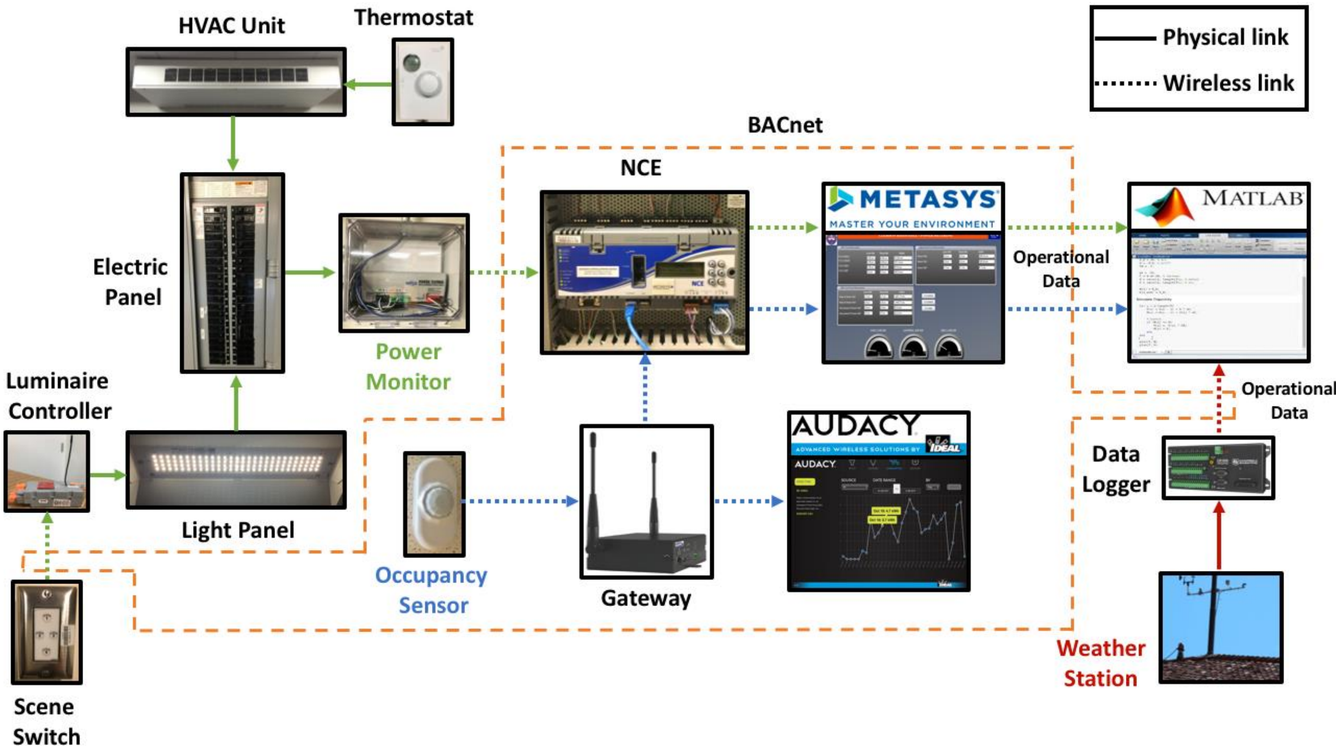

Fig. 2. Integration of all the IoT devices and technologies and software platforms using BACnet protocol for data measurement in RTPIS laboratory

with four custom scene settings/light levels (0, 30%, 60%, and 100%) for each room, which allowed on-site occupants to instantly adjust the light levels according to their preference. The scene switches operate in highly reliable 915 MHz spectrum, easy to install, and wireless.

RTPIS laboratory's occupancy state was measured using the Audacy ceiling-mount occupancy sensors [11]. Additionally, the occupancy sensors controlled the light panels switching on and off by communicating the occupancy state of the rooms in the RTPIS laboratory with the luminaire controllers over BACnet protocol. These sensors offer the following benefits:

- Infrared devices capable of detecting occupancy and/or vacancy
- BACnet compatible
- Easy to mount

The central processing hub included the Audacy gateway [12] and the Johnson Controls Network Control Engine (NCE) [13]. Operational data from occupancy sensors, weather station, and power meters is wirelessly transmitted to the workstation using the central processing hub, where the data is processed to generate actionable information. This actionable information is wirelessly transmitted to thermostats and smart luminaire controllers using the central processing hub. All communications take place over the BACnet protocol.

2) *Software Platforms*

The software platforms used to manage and drive the IoT devices deployed in the RTPIS laboratory include Metasys Building Automation System software, online Audacy Interface, and MATLAB (Fig. 2). A brief description of each software platform is provided below.

Metasys Building Automation System software [14] by Johnson Controls is the state-of-the-art software platform for modern building energy management. It is a world-class, intelligent technology system that connects the electric loads and IoT devices in the RTPIS laboratory, enabling them to communicate the desired information that makes their optimized control possible. This results in increased energy efficiency and enhanced occupant's comfort, safety, and productivity. The following are some of the features and benefits of Metasys Building Automation software:

- Web accessible with full capabilities from laptops, tablets, and smart phones
- Easy space-based navigation
- Intuitive design that reduces learning time
- Shows space-by-space status info that makes troubleshooting relatively easy
- Easy to compare equipment performances though single-view equipment summary
- Increased interoperability through support for BACnet protocol
- Follows government and industry best practices for continuous security improvements

Online Audacy Interface [15] provides a convenient access, monitoring, and control platform for wireless lighting control using Audacy IoT devices. It comprises of three primary sections viz. IoT device upload, IoT device scheduling and control, and electric energy consumption profile. Audacy IoT devices can be uploaded to the online interface during installation from the IoT device upload section via the mobile app or desktop computer. Once the IoT devices are uploaded and activated, their preferred control and scheduling settings can be instantly configured from the IoT device control section to achieve the desired light intensity levels across the building environment. Sliders are available in the user interface of the IoT device scheduling and control section to set dim levels and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2975847, IEEE Internet of Things Journal

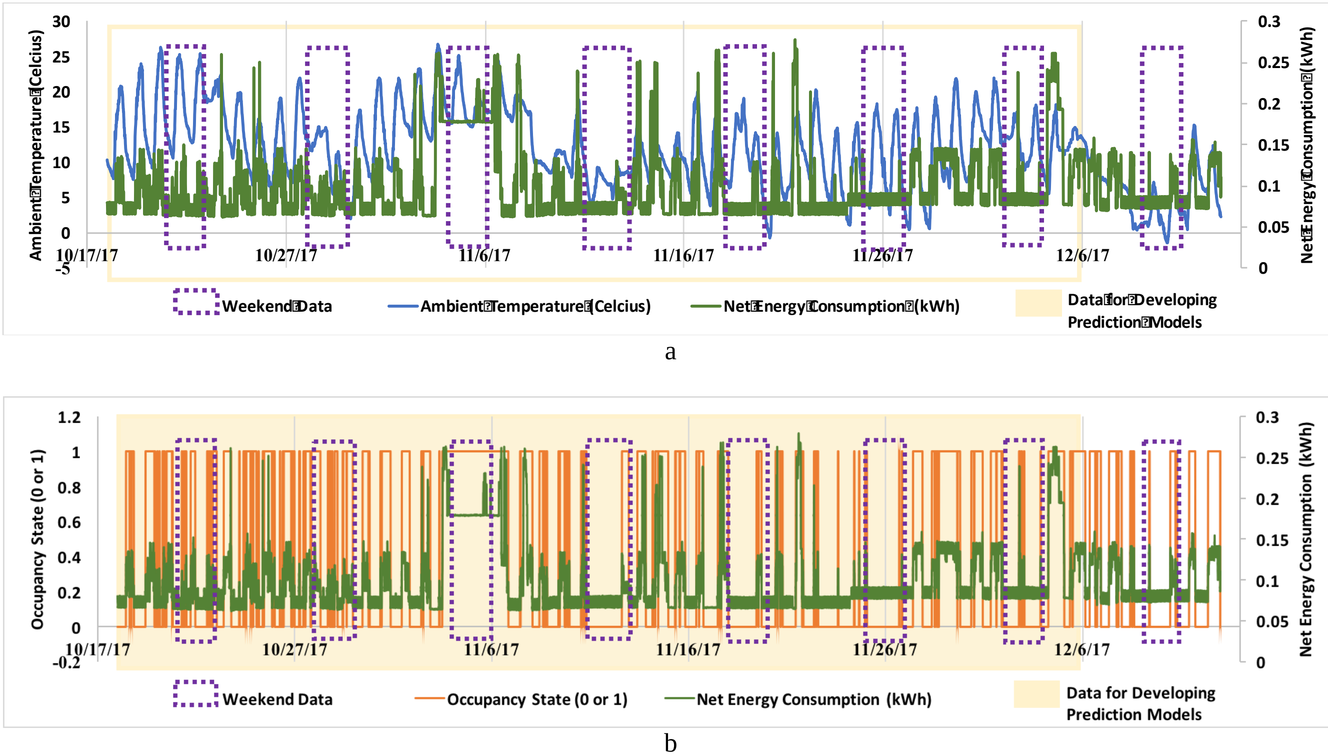

Fig. 3. (a) Temperature and (b) occupancy data vs. net energy consumption data for October 18 – December 12, 2017

decide timeout delays for the light panels. The lighting load energy consumption profiles can be visualized from the electric energy consumption section. Energy consumption profiles can be broken down by date, room, range, or time period and can be easily exported for performing further analytics. The following are some of the features and benefits of Online Audacy Interface:

- Easy to program and customize
- Flexible scheduling
- Accessible from Apple devices through the mobile app
- Supports BACnet protocol

MATLAB or matrix laboratory [16] is a proprietary programming language that was developed by MathWorks. It is the state-of-the-art multi-paradigm numerical computing environment with professional mathematical, graphical, and programming toolboxes and interactive apps that are scalable to run on clusters, GPUs, and clouds. In this research, MATLAB was used to analyze operational data from the IoT devices and develop optimization algorithms to create optimized control models for the IoT actuators controlling the electric loads in the RTPIS laboratory.

3) BACnet Protocol

BACnet protocol [17] is an internationally recognized and accepted protocol. It is an American national standard, European standard, national standard in more than 30 countries, and ISO global standard. BACnet protocol was developed by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). Its increasing use is attributed to the following benefits:

- Single operator workstation for all systems
- Competitive System expansion
- Eliminates fear of being “locked in”
- Possibility of integrating all building automation and control functions
- Interoperability to facilitate efficient data sharing, alarm and event management, scheduling, and remote device and network management

4) Data Measurement

The data sources for this research include a real-time weather station (installed on the roof of Riggs Hall at Clemson University) for ambient temperature data, occupancy sensors for occupancy state data, and smart power meters for electric energy consumption data. A circuit diagram showing the integration of all the IoT devices and technologies and software platforms using BACnet protocol for data measurement is shown in Fig. 2. The power monitor measures the electric energy consumption from the HVAC units and light panels through the electric panel. This measured data is transmitted to the NCE. The NCE transmits the electric energy consumption data to the Metasys software, where it is then exported to MATLAB. The occupancy sensor measures the occupancy state, which is transmitted to the NCE via Audacy gateway. The NCE transmits the occupancy state data to the Metasys software from where it is exported to MATLAB. The temperature sensor on the weather station measures the ambient temperature, which is exported to MATLAB via a data logger. All the measured data exported to MATLAB is the operational data that is employed for developing the electric energy consumption prediction model. All data transmission takes place over the BACnet protocol.

The ambient temperature, occupancy state, and electric energy consumption data was measured over a period of eight weeks (October 18 – December 12, 2017) and had a resolution of 10 minutes per utility industry standard, which makes it a total of 8064 data points (Fig. 3). This measured data was divided into two categories: (1) data used to develop the electric energy prediction models and (2) data used to test the prediction accuracy of the developed models. Seven weeks of data (October 18 – December 5, 2017) (7056 data points) was used to develop both Elman RNN and exponential models for electric energy consumption prediction. This data was further separated based on weekday data (thirty-five weekdays = 5040 data points) and weekend data (fourteen weekend days = 2016 data points) for improving the accuracy of electric energy consumption predictions as trying to predict electric energy consumption for weekend using weekday data or vice versa is not ideal. The remaining one week of data (December 6 – December 12, 2017) (1008 data points) was used to test the developed models' electric energy consumption prediction accuracies.

III. Electric Energy Consumption Prediction

A. Problem Definition

The problem definition (Fig. 4) for this study involves measuring historic data for ambient temperature, occupancy state, and electric energy consumption over a period of seven weeks (October 18 – December 5, 2017) and utilizing this data to build both Elman RNN and exponential prediction models that recognize a relationship between the net electric energy consumption by the electric loads (HVAC units and light panels) in the RTPIS laboratory and the laboratory's ambient temperature and occupancy state. These models can be used for real-time and near future electric energy consumption estimation and prediction based on the forecasted ambient temperature data and scheduled building occupancy state data. For validating the models, the electric energy consumption was predicted for the period December 6 – 12, 2017 and compared with the measured (actual) electric energy consumption data for the specified period.

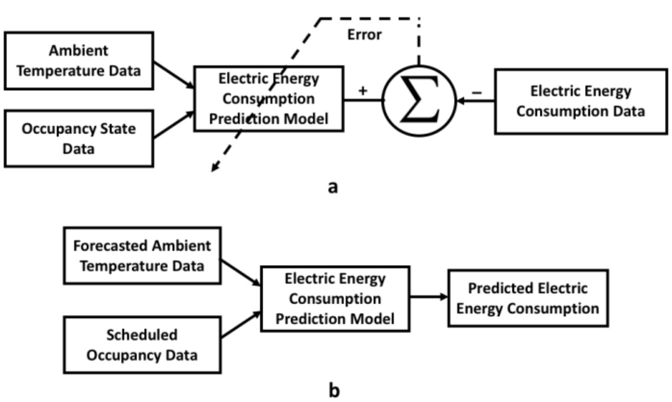

Fig. 4. Problem definition. (a) Electric energy consumption prediction model under development and (b) developed model used in prediction mode (adapted from [5])

B. Elman Recurrent Neural Network Model

Elman RNN was conceived and first used by Jeff Elman in 1990 [18]. It is usually a two-layer backpropagation network where the output of the hidden layer in the previous time step (*k-1*) is fed back as an input to the hidden layer in the current time step (*k*). This feedback is the recurrent connection in the Elman RNN that gives the neural network a short term memory that allows it to generate and recognize spatial and temporal patterns [19, 20]. The hidden/recurrent layer in an Elman RNN has a hyperbolic tangent sigmoid (*tansig*) or log-sigmoid (*logsig*) transfer function whereas the output layer has a linear (*purelin*) transfer function [19].

The Matlab function used to create an Elman RNN is *newelm* [19, 21]. The default parameter settings when using *newelm* include Nguyen-Widrow layer initialization (*initnw*) function for initializing the weights and biases of each layer, *tansig* and *purelin* transfer functions for hidden and output layers respectively, BFGS quasi-Newton backpropagation (*trainbfg*) function for backpropagation training, gradient descent with momentum weight and bias learning (*learngdm*) function for backpropagation weight/bias learning, and mean squared normalized error performance (*mse*) function for performance. The mathematical equations for *tansig* (1), *purelin* (2), and *mse* (3) functions are as follows [16]:

$$tansig(n) = \frac{2}{(1+e^{-2n})} - 1 \quad (1)$$

$$purelin(n) = n \quad (2)$$

$$mse = \frac{1}{N}\sum_{i=1}^{N}(error_i)^2 \quad (3)$$

There are two Matlab functions that can be used for training Elman RNN viz. *train or adapt* [19]. The following happens at each epoch when using the *train* function: The network is presented with the input sequence for which the outputs are calculated. Next, an error sequence is generated by comparing the outputs with the target sequence. Once the error is known, error gradients for each bias and weight are determined by backpropagating this error sequence. It is important to note here that the value of the gradients is an approximation since in Elman RNN the biases and weights contributions to error through the delayed feedback or recurrent connection are ignored. Finally, the weights are updated using the approximate gradient with a backpropagation training function. The gradient descent with momentum and adaptive learning rate backpropagation (*traingdx*) function is recommended and generally used. In the case of the *adapt* function, the following happens at each time step: The network is presented with the input vectors and an error is generated. Next, error gradients for each bias and weight are determined by backpropagating this error. As with the train function, the value of the gradients in this case is an approximation. Finally, the weights are updated using the approximate gradient with a learning function. The gradient descent with momentum weight and bias learning (*learngdm*) function is recommended and generally used. Since the training and adaption in an Elman RNN occurs using the error gradient approximation, this network is not as reliable as some other kinds of networks. A solution to overcome this drawback and give an Elman RNN the best chance at learning and solving a problem is to have more neurons in the hidden layer than is typical for any other network to solve a similar problem [16, 19].

A two-layer Elman RNN model was developed and used for electric energy predictions. It comprised of two input neurons, ten hidden layer neurons, and a single output layer neuron (Fig. 5). For training the Elman RNN model, the measured weekday and weekend training data (described above) was divided into three datasets viz. the input, target, and sample dataset. The input dataset included the ambient temperature and occupancy state data, the target dataset included the measured electric energy consumption data, and the sample dataset included the forecasted temperature and scheduled occupancy state data for electric energy consumption prediction. To avoid overfitting, the data in each dataset was randomly divided into three subsets: 70% of the data was used for training, 15% for validation, and 15% for testing.

TABLE I
ELMAN RNN TRAINING PARAMETERS

Parameter	MATLAB Implementation	Value
Maximum number of epochs to train	*net.trainParam.epochs*	1000
Performance goal	*net.trainParam.goal*	0
Learning rate	*net.trainParam.lr*	0.01
Ratio to increase learning rate	*net.trainParam.lr_inc*	1.05
Ratio to decrease learning rate	*net.trainParam.lr_dec*	0.7
Maximum validation failures	*net.trainParam.max_fail*	1000
Maximum performance increase	*net.trainParam.max_perf_inc*	1.04
Momentum constant	*net.trainParam.mc*	0.9
Minimum performance gradient	*net.trainParam.min_grad*	10^{-5}
Maximum time to train in seconds	*net.trainParam.time*	infinite

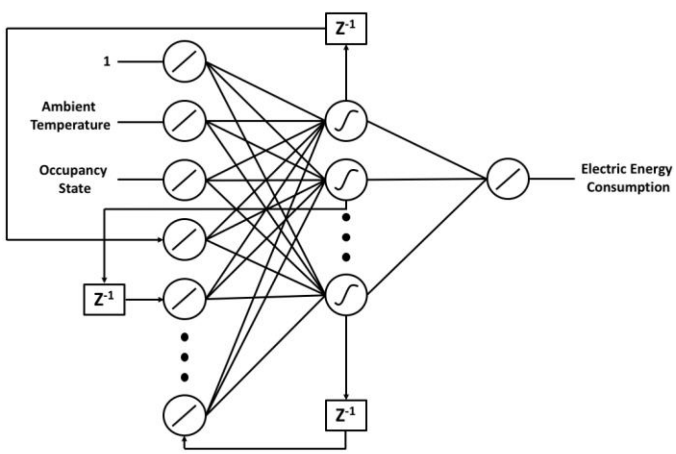

Fig. 5. Elman RNN model for electric energy consumption prediction

Table I shows the parameters used for training the Elman RNN. Initially, the specified parameters were set to the standard, default values, which were then further refined by trial-and-error until the desired network performance was achieved. The network was trained separately for weekday and weekend data to obtain the corresponding weights and biases.

The performance (*perf*) derivatives with respect to the bias and weight variables X were calculated using backpropagation. The gradient descent with momentum is used to adjust each variable (4) [16].

$$dX = mc * dX_{prev} + lr * mc * \frac{dperf}{dX} \tag{4}$$

where,
dX_{prev} = previous change to bias or weight
mc = momentum constant
lr = learning rate

After each epoch, the network *perf* is evaluated and the learning rate (*lr*) is updated accordingly. If the *perf* decreases towards the goal, *lr* is increased by *lr_inc*. If the *perf* increases beyond the *max_perf_inc* value, *lr* is decreased by *lr_dec*. The Elman RNN training stops when any of the following conditions hold true [19]:

- Maximum number of epochs is reached
- Maximum time to train is exceeded
- Performance goal is achieved
- Performance gradient falls below *min_grad* value
- Validation performance increases more than *max_fail* times since the last time it decreased

The developed Elman RNN model was used to generate the RTPIS laboratory's electric energy consumption predictions for the period December 6 – 12, 2017 using the forecasted temperature and scheduled occupancy state data. The electric energy consumption for the specified period was predicted for weekdays and weekends, separately using their respective weights and biases. The predicted values were then compared with the measured net electric energy consumption values for these days and the $\%\,Error$ was calculated using (5). The results are shown and discussed in Section IV.

$$\%\,Error = \left(\frac{|Predicted\ Energy\ Consumption - Measured\ Energy\ Consumption|}{Measured\ Energy\ Consumption}\right) X100 \tag{5}$$

C. Exponential Model

The authors in [5] developed an exponential model for electric energy consumption prediction in a smart building environment. The objective function of the exponential model was defined as follows ((6) – (8)) [5]:

$$P_{net} = A(Temp) + B((Temp)^2) + C\left(e^{-(Temp)}\right) + D(occupy) + E((occupy)^2) + F\left(e^{-(occupy)}\right) + G \tag{6}$$

where,
P_{net}= instantaneous net RTPIS laboratory power consumption (kW)
$Temp$ = ambient temperature (°C)
$occupy$= occupancy state of the RTPIS laboratory
$A, B, and\ C$= decision variables for ambient temperature data having units of kW/°C, kW/°C^2, and kW, respectively
$D, E, and\ F$=decision variables for occupancy state data, all having units of kW
G=constant (kW)

The authors in [5] obtained the values of the decision variables and constant using the particle swarm optimization (PSO) algorithm [22, 23]. However, limitations exist including

(1) measurement data was shallow (only eight days) and (2) data was not categorized into weekday and weekend data. These limitations are addressed in this paper by using a much larger dataset that is categorized into weekday and weekend data (described above). Based on the new data, the values of the decision variables and constant were updated using the PSO algorithm and tabulated for weekday data and for weekend data (Table II). The parameters specified in the PSO algorithm are listed below [5]:

- Number of particles = 20
- Number of iterations = 500
- Number of dimensions = 7
- Particle velocity and position range = [-100, 100]
- Inertia weight (w) = 0.729
- Acceleration constants $c_1 = c_2 = 1.49$

TABLE II
VALUE OF THE DECISION VARIABLES, CONSTANT, AND MINIMUM ERROR FOR WEEKDAY AND WEEKEND DATA

Parameter	Value for Weekday Data	Value for Weekend Data
A (kW/°C)	0.0022	0.0382
B (kW/°C^2)	-0.0002	-0.0027
C (kW)	0.3244	-0.0806
D (kW)	0.2291	0.1350
E (kW)	0.1407	0.1275
F (kW)	0.2342	0.1200
G (kW)	0.2459	0.2764
Minimum Error (gbest)	98.1959	74.4046

Once the value of the decision variables was determined, P_{net} was calculated using (6). Subsequently, the net energy consumption of the RTPIS laboratory over time T was calculated using equation (7).

$$E_{netc} = \int_{t=1}^{T} P_{net}\, dt \tag{7}$$

where,
E_{netc}= calculated net energy consumption (kWh)
Using (6) and (7), electric energy consumption was predicted for the period December 6 – 12, 2017 using the same forecasted temperature and scheduled occupancy state data as used in the Elman RNN model. The electric energy consumption for the specified period was predicted for weekdays and weekends, separately using their respective parameters. The predicted values were compared with the measured net energy consumption values for the period, and the $\%\,Error$ was calculated using (8). The results are shown and discussed in Section IV.

$$\%\,Error = \left(\frac{|E_{netc} - E_{netm}|}{E_{netm}}\right) X100 \tag{8}$$

where,
E_{netm}= measured net energy consumption (kWh)

Some of the important equations used in the PSO algorithm are as follows [27]:
A particle i in a swarm is represented by its current position (x_i), current velocity (v_i), and personal best position (y_i). If f denotes the objective function, then the personal best position of a particle i at time k is updated according to (9).

$$y_i(k+1) = \begin{cases} y_i(k) & if\ f(x_i(k+1)) \geq f(y_i(k)) \\ x_i(k+1) & if\ f(x_i(k+1)) < f(y_i(k)) \end{cases} \tag{9}$$

The global best particle's position vector ($\hat{y}$) is given by (10).

$$\hat{y}(k) \in \{y_0, y_1, \dots, y_s\} = \min\{f(y_0(k)), f(y_1(k)), \dots, f(y_s(k))\} \tag{10}$$

where,
s = size of swarm
The velocity v_i of a particle i is updated according to (11).

$$v_{i,j}(k+1) = wv_{i,j}(k) + c_1 r_{1,j}(k)\left(y_{i,j}(k) - x_{i,j}(k)\right) + c_2 r_{2,j}(k)\left(\hat{y}_j(k) - x_{i,j}(k)\right) \tag{11}$$

where,
$j \in 1, \dots, N_d$
N_d= dimension of the problem
$v_{i,j}$= j-th element of the velocity vector of the i-th particle
w= inertia weight
c_1 and c_2= acceleration constants
$r_{1,j}, r_{2,j} \sim U(0,1)$
The position x_i of a particle i is updated according to (12).

$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{12}$$

IV. EXPERIMENT RESULTS AND DISCUSSIONS

A. Prediction Results

The comparison of the predicted and the measured net electric energy consumption values using Elman RNN and exponential models, along with the % *Error* values are tabulated in Table III and shown in Fig. 6. From the small % *Error* values, it can be inferred that the predicted and measured net electric energy consumption values are very similar for all the days. This validates the developed Elman RNN and exponential models and shows they work well for electric energy consumption prediction in a smart building environment.

From the comparison between the Elman RNN model and the exponential model, it is clear that the Elman RNN model outperforms the exponential model on six out of seven days of electric energy consumption predictions, thereby making it a more efficient approach for real-time and near future electric energy consumption estimation and prediction in an IoT driven building environment.

B. Application of Developed Prediction Models

The developed prediction models can be used to inform the operation of a computational systems thinking machine (CSTM). This CSTM is currently under development in the RTPIS laboratory and is referred to as the intelligent computational engine (ICE) (Fig. 7). The ICE will serve as the thought and action center of a building environment. The various features and impact of the ICE are as follows. As shown in Fig. 7, the ICE is a computational, information and action engine that receives operational data from IoT sensors, processes it, and generates actionable information for the IoT actuators. This information can be accessed using user interfaces such as a smartphone, tablet, or personal computer. The ICE can also control the switching of the electric power from the electric grid to supply energy to a building's electric loads on an as needed basis [5].

TABLE III
PREDICTED ENERGY CONSUMPTION AND MEASURED NET ENERGY CONSUMPTION USING ELMAN RNN AND EXPONENTIAL MODELS

Date	Measured Net Energy Consumption (kWh)	Predicted Energy Consumption with Elman RNN Model (kWh)	Predicted Energy Consumption with Exponential Model (kWh)	%Error with Elman RNN Model	% Error with Exponential Model
December 6, 2017	16.7333	16.2696	15.0489	2.7709	10.0656
December 7, 2017	15.4564	15.4205	14.6196	0.2323	5.4135
December 8, 2017	13.1197	12.4325	14.1332	5.2381	7.7244
December 9, 2017	12.1032	11.8417	11.2118	2.1609	7.3650
December 10, 2017	12.8876	12.6478	11.1339	1.8606	13.6083
December 11, 2017	14.0726	13.5625	13.5831	3.6246	3.4781
December 12, 2017	16.2459	15.9942	15.0152	1.5496	7.5755

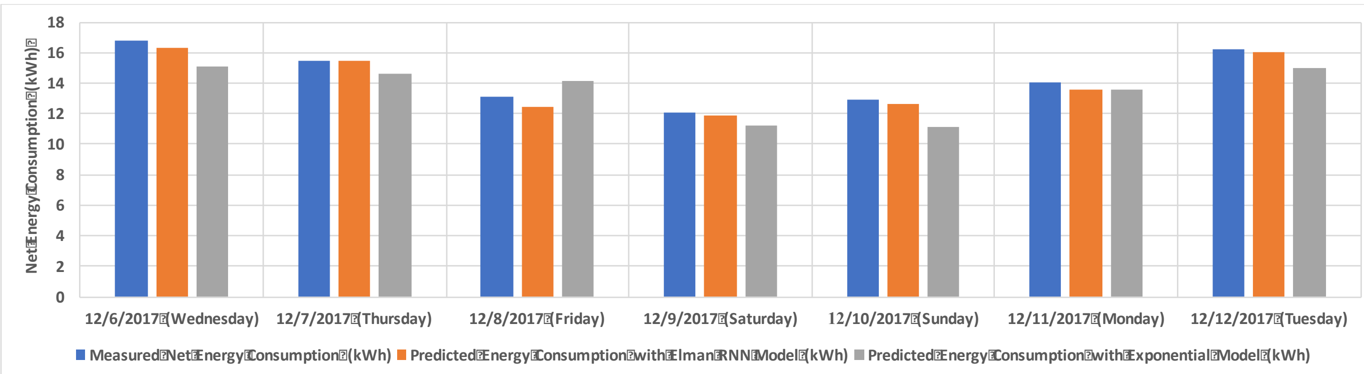

Fig. 6. Predicted and the measured net energy consumption values using Elman RNN and exponential models

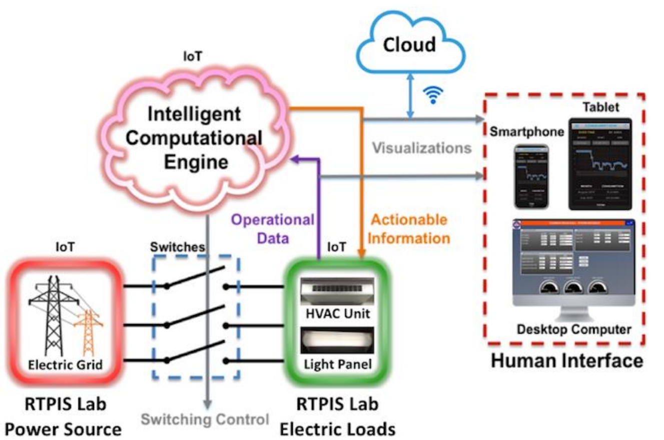

Fig. 7. Intelligent Computational Engine with IoT technologies integration [5]

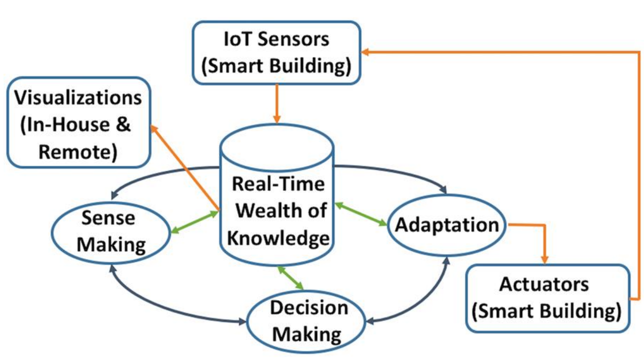

Fig. 8. Computational system for smart building (extended from [25])

The general control framework of ICE for efficient energy management in an IoT driven building environment consists of sensing, communication, computation, control, and visualization blocks. The sensing block includes IoT sensors that record operational data (e.g., energy consumption, temperature, occupancy, etc.) and communicate it to the computation block. The computation block uses historic operational data to develop a prediction model/algorithm, which is used for analyzing forecasted/real-time operational data to generate energy consumption prediction values. The prediction values are communicated to the control block, which includes a control model/algorithm that uses the predicted energy value as a reference to compare the measured energy value (obtained from IoT sensors) to generate optimized control parameters. These parameters are communicated to the IoT actuators, which optimally regulate the operation of the electric loads to reduce energy waste (improve energy efficiency). The IoT sensor data and energy consumption prediction values can be visualized, in-house and remote, using the visualization block. The communication block facilitates data exchange between all the blocks.

The general objective function for generating control parameters for optimized control of the electric loads in a smart building environment is given by (13) [24] subject to the constraint (14).

$$max(\sum_{i=1}^{N} P_i . L_i) \quad (13)$$

where P_i is the electric load priority weighting and L_i is the electric load magnitude (kWh) for a particular electric load i. N represents the total number of electric loads.

$$MEC \leq SF \times Ref \quad (14)$$

where *MEC* is the measured/actual energy consumption, *SF* is the safety factor, and *Ref* is the reference to guide the optimization procedure.

The impact of the developed ICE includes minimized wasted energy, improved system efficiency, reduced energy costs, increased energy savings, enhanced user comfort, increased return on investment, and reduced carbon emissions.

The ICE when applied to a building environment can transform it to a smart building environment. A smart building environment comprises of a computational system (Fig. 8) which includes IoT sensors and actuators, wireless communication technologies, control systems, and visualizations. This imparts adaptation (control), sense-making (communication), and decision-making (computation) capabilities to various electronic devices present in the building [25, 26]. Examples of existing intelligent energy management systems proposed in literature include the intelligent cloud home energy management system [27] and the intelligent energy distribution management system [28].

V. Conclusions

The importance of IoT deployment in a building environment for efficient energy management was presented in this paper. This work was demonstrated in a real working environment. An Elman recurrent neural network model and exponential model were developed and applied to the RTPIS laboratory for real-time and near future electric energy consumption estimation and prediction. Having prediction capability is beneficial to reduce energy waste by generating and supplying electric energy as predicted to meet the electric load demands. This, in turn, results in reduced carbon emissions and reduced costs. Although developed for the RTPIS laboratory at Clemson University, the Elman RNN and exponential models are scalable and flexible, providing the capability to adapt these models for usage with any IoT driven building environment as conceptualized in [29-31]. As a future work, the developed electric energy consumption prediction models will be employed for the development and application of the Intelligent Computational Engine in the RTPIS laboratory for achieving automated, real-time, and optimized control of electric energy consumption to minimize inefficient energy management, wasted energy resources, and expensive energy costs.

References

[1] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta and M. Alikarar, "A smart home energy management system using IoT and big data analytics approach," in IEEE Transactions on Consumer Electronics, vol. 63, no. 4, pp. 426-434, November 2017.

[2] A. C. Luna, N. L. Diaz, M. Graells, J. C. Vasquez and J. M. Guerrero, "Cooperative energy management for a cluster of households prosumers," in IEEE Transactions on Consumer Electronics, vol. 62, no. 3, pp. 235-242, August 2016.

[3] J. Han, C. S. Choi, W. K. Park, I. Lee and S. H. Kim, "Smart home energy management system including renewable energy based on ZigBee and PLC," in IEEE Transactions on Consumer Electronics, vol. 60, no. 2, pp. 198-202, May 2014.

[4] Y. S. Son, T. Pulkkinen, K. D. Moon and C. Kim, "Home energy management system based on power line communication," in IEEE Transactions on Consumer Electronics, vol. 56, no. 3, pp. 1380-1386, Aug. 2010.

[5] G. Bedi, G. K. Venayagamoorthy and R. Singh, "Pattern recognition for electric energy consumption prediction in a laboratory environment," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 2017, pp. 1-8.

[6] Real-Time Power and Intelligent Systems (RTPIS) Laboratory, http://rtpis.org, last accessed 08/04/2017

[7] Power Patrol, https://www.setra.com/products/power-monitoring/power-patrol-revenue-grade-power-meter, last accessed 08/04/2017

[8] Pneumatic to Direct Digital Control DDC Room Thermostats, http://www.johnsoncontrols.com/buildings/hvac-controls/thermostats/pneumatic-to-direct-digital-control-ddc-room-thermostats, last accessed 08/04/2017

[9] Luminaire Controller, Internal-Mount, http://www.audacywireless.com/audacy/products/lighting-controls/luminaire-controllers/scl-1000.aspx, last accessed 08/04/2017

[10] Scene Switch, http://www.audacywireless.com/audacy/products/lighting-controls/switches/wss-1200.aspx, last accessed 11/08/2017

[11] Ceiling-Mount Motion Sensor, http://www.audacywireless.com/audacy/products/lighting-controls/motion-sensors/vsc-1300.aspx, last accessed 08/04/2017

[12] Gateway, http://www.audacywireless.com/audacy/products/lighting-controls/gateway/gw-1100.aspx, last accessed 08/04/2017

[13] Network Control Engine Catalog Page, http://cgproducts.johnsoncontrols.com/CAT_PDF/1900455.pdf, last accessed 08/04/2017

[14] Metasys Software, http://www.johnsoncontrols.com/buildings/building-management/building-automation-systems-bas/metasys-software-and-servers, last accessed 11/08/2017

[15] Online Audacy Interface, http://www.audacywireless.com/online-interface/, last accessed 11/08/2017

[16] Matlab, https://www.mathworks.com/products/matlab.html, last accessed 11/08/2017

[17] BACnet, http://www.bacnet.org, last accessed 08/04/2017

[18] A J. L. Elman, Finding Structure in Time, Cognitive Science 14, 179-211, 1990.

[19] Neural Network Toolbox for Use with MATLAB, by H. Demuth and M. Beale, The MathWorks, Inc, 2002.

[20] A Field Guide to Dynamical Recurrent Networks, by J. F. Kolen and S. C. Kremer, The Institute of Electrical and Electronic Engineers, Inc., New York, 2001.

[21] L. Ren, Y. Liu, Z. Rui, H. Li and R. Feng, Application of Elman Neural Network and MATLAB to Load Forecasting, 2009 International Conference on Information Technology and Computer Science, Kiev, 2009, pp. 55-59.

[22] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez and R. G. Harley, Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems, in IEEE Transactions on Evolutionary Computation, vol. 12, no. 2, pp. 171-195, April 2008.

[23] A. Abraham, C. Grosan, and V. Ramos, Swarm Intelligence in Data Mining, Studies in Computational Intelligence, Volume 34, Springer-Verlag Berlin Heidelberg 2006.

[24] P. Mitra and G. K. Venayagamoorthy, "Implementation of an Intelligent Reconfiguration Algorithm for an Electric Ship's Power System," in IEEE Transactions on Industry Applications, vol. 47, no. 5, pp. 2292-2300, Sept.-Oct. 2011.

[25] G. K. Venayagamoorthy, "Dynamic, Stochastic, Computational, and Scalable Technologies for Smart Grids," in IEEE Computational Intelligence Magazine, vol. 6, no. 3, pp. 22-35, Aug. 2011.

[26] G. Bedi, G. K. Venayagamoorthy and R. Singh, " Internet of Things (IoT) Sensors for Smart Home Electric Energy Usage Management," 2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Galle, Sri Lanka, pp. 1-6, 2016.

[27] J. Byun, I. Hong and S. Park, "Intelligent cloud home energy management system using household appliance priority based scheduling based on prediction of renewable energy capability," in IEEE Transactions on Consumer Electronics, vol. 58, no. 4, pp. 1194-1201, November 2012.

[28] I. Hong, B. Kang and S. Park, "Design and implementation of intelligent energy distribution management with photovoltaic system," in IEEE Transactions on Consumer Electronics, vol. 58, no. 2, pp. 340-346, May 2012.

[29] Roberto Casadei, Giancarlo Fortino, Danilo Pianini, Wilma Russo, Claudio Savaglio, Mirko Viroli: Modelling and simulation of Opportunistic IoT Services with Aggregate Computing. Future Generation Comp. Syst. 91: 252-262 (2019)

[30] Giancarlo Fortino, Wilma Russo, Claudio Savaglio, Weiming Shen, Mengchu Zhou: Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation. IEEE Trans. Systems, Man, and Cybernetics: Systems 48(11): 1939-1956 (2018)

[31] Giancarlo Fortino, Raffaele Gravina, Wilma Russo, Claudio Savaglio: Modeling and Simulating Internet-of-Things Systems: A Hybrid Agent-Oriented Approach. Computing in Science and Engineering 19(5): 68-76 (2017)

Originally from India, **Guneet Bedi** is a Ph.D. graduate from the Department of Electrical and Computer Engineering (ECE) at Clemson University, where he researched Internet of Things applications in Electric Power and Energy Systems. Guneet received his B.E. in Electronics and Telecommunication from University of Pune, India (2011) and his M.S. in Electrical Engineering from Clemson University (2014). While in the ECE program, Guneet worked as a graduate research, teaching, instructional, and grading assistant as well as a mentor for K-12 and undergraduate science enrichment programs. Guneet has published several peer-reviewed papers and has presented his research at a number of conferences. He has also served as a reviewer for IEEE Internet of Things journal and Elsevier Computers and Security journal. Besides his academic pursuits, Guneet has served as the President of Graduate Student Government, President of International Student Association, and the Vice-President of Clemson Indian Students' Association. Guneet played on Clemson's water polo club team for two successive years and is a graduate student member of the Omicron Delta Kappa national level leadership honor society and the Institute of Electrical and Electronics Engineers. During his time at Clemson, Guneet has been honored with awards for his outstanding scholarship, leadership, teaching, and service.

G. Kumar Venayagamoorthy is currently the Duke Energy Distinguished Professor of Power Engineering and a Professor of Electrical and Computer Engineering and Automotive Engineering with Clemson University, Clemson, SC, USA. He is also the Founder and Director of the Real-Time Power and Intelligent Systems Laboratory (http://rtpis.org) with Clemson University. He is an Honorary Professor with the School of Engineering, University of KwaZulu–Natal, Durban, South Africa. Dr. Venayagamoorthy's interests are in the research, development and innovation of advanced computational methods for smart grid operations, including intelligent sensing and monitoring, power system optimization, stability and control, and signal processing. He has published ~ 500 refereed technical articles. His publications are cited ~14,000 times with a *h*-index of 59. Dr. Venayagamoorthy has been involved ~ 70 sponsored projects in excess of $10 million. He has received several awards from IEEE and other professional societies and institutions for his contributions to field of electrical and electronic engineering.

Dr. Venayagamoorthy is involved in the leadership and organization of many conferences including the General Chair of the Power System Conference (Clemson, SC, USA) since 2013, and Pioneer and Chair/co-Chair of the IEEE Symposium of Computational Intelligence Applications in Smart Grid (CIASG) since 2011. He is currently the Chair of the IEEE PES Working Group on Intelligent Control Systems, and the Founder and Chair of IEEE Computational Intelligence Society (CIS) Task Force on Smart Grid. Dr. Venayagamoorthy is a Senior Member of the IEEE and International Neural Network Society, and a Fellow of the IET (UK), and the SAIEE (South Africa). He has served as an Editor/Guest Editor of several IEEE and Elsevier journal.

Rajendra Singh is D. Houser Banks professor in the Holcombe Department of Electrical and Computer Engineering and Automotive Engineering and Executive Director of Sustainable Electrification of Transportation Minor Program at Clemson University. With proven success in operations, project/program leadership, R&D, product/process commercialization, and start-ups, Dr. Singh is a leading technologist with four decades of industrial and academic experience of photovoltaic and semiconductor industries. His current research interest is to provide disruptive and transformative new business opportunities in power, transportation, and water sectors based on local power network of photovoltaics and batteries. Dr. Singh has received a number of national and international awards. On April 17, 2014 he was honored by US President Barack Obama as a White House "Champion of Change for Solar Deployment" for his leadership in advancing solar energy with Photovoltaics technology. On January 25, 2019 he was honored with the prestigious Hind Rattan Award presented by Dr. Bibek Debroy, Chairman of the Economic Advisory Council to Prime Minister of India Mr. Narendra Modi. Dr. Singh is Fellow of IEEE, SPIE, AAAS and ASM.