WSN-Based Data-Driven Digital Twin for Energy Efficient HVAC Systems

Mahmud Alosta and Saad Abobkr
System Engineering Department
École de technologie supérieure (ÉTS)
University of Québec, Montréal, Canada
{mahmud.alosta, saad.abobakr}@etsmtl.ca

Amine El Kaouachi
Quantolio Technologies Inc
Montréal, Canada
amine.elkaouachi@quantolio.com

Lokman Sboui
System Engineering Department
École de technologie supérieure (ÉTS)
University of Québec, Montréal, Canada
lokman.sboui@etsmtl.ca

***Abstract*—Efficiently managing of energy consumption while ensuring a comfortable indoor environment is a crucial challenge in Heating Ventilation and Air Condition (HVAC) system control. Traditional approaches rely on physical models, while effective to some extent, entail laborious, time-consuming processes and lack scalability, imposing a substantial burden on HVAC system controllers. In this paper, we propose a data driven approach combined with Wireless Sensor Network (WSN) and Digital Twins (DT), allowing for the representation of system components in both the physical and virtual worlds. More specifically, a Long Short-Term Memory (LSTM) is integrated to enable in-advance modeling and prediction of indoor temperature for a potential Model Predictive Controller (MPC). The LSTM model is trained and validated using a dataset collected from sensors deployed in a candidate building. Several benchmarks were evaluated to ensure the efficiency of the proposed system model.**

***Index Terms*—Digital Twin, LSTM, HVAC, Time Series**

I. Introduction

Buildings are responsible for a considerable fraction of the energy consumption in both the United States and Canada, with figures of 40% and 30 %, respectively, of the total energy consumption. Due to extreme weather conditions, it is important for Canadians to have reliable and efficient HVAC systems to ensure a comfortable indoor conditions. Therefore, HVAC systems account for approximately 60% of the energy consumption in commercial and institutional buildings and 40% in residential buildings [1] [2].

Both research and industrial communities have been actively pursuing strategies to optimize the energy consumption of HVAC systems. These efforts are centered on the development of model-based control solutions. Mainly, two main paths of model-based HVAC control research have been observed.

First, physics-based models (white box) such as resistance capacitance (RC) models and simulation tools such as TRNSYS and EnergyPlus [3]–[6]. These models are derived from the principles of thermodynamics related to buildings and explicitly incorporate the modeling of heat transfer between various building components. Furthermore, they offer the advantage of providing a detailed and precise representation of temperature variations. However, the development of such models involves large number of parameters for detailed understanding of system physics, which enlarges their dimensionality, and, consequently, makes them computationally expensive [1] [7]. This challenge becomes a significant focal point, particularly when considering the deployment of these models at the edge layer, where devices are characterized by limited resources.

Secondly, data-driven models (black box) have gained acceptance for addressing physics-based model limitations. They enable dynamic behavior simulation through the correlation between input and output variables. Such spectrum of models include the autoregressive moving average (ARIMA), artificial neural network (ANN), and support vector machine (SVM) [1] [10]. These models are characterized by their ability to simulate HVAC system behavior using only HVAC-related data, requiring less comprehensive knowledge compared to white-box models. This, in turn allows them to be flexible and scalable. Furthermore, the structure of a data-driven model can be adjusted to meet the specific prediction accuracy needs of different buildings [9].

In this paper, we propose the LSTM (Long-Short Term Memory) neural network model to establish an indoor temperature prediction model. The used data were collected from an actual residential building and the model was developed to generate precise predictions conceived as an input to a Model predictive controller (MPC). Furthermore, since a real building is involved in this work, we consider the concept of digital twins (DT) to model the functionalties and data interaction of the proposed system. As defined by [10] ”The DT represents and reflects its physical twin and remains its virtual counterpart across the object's entire lifecycle”. The novelty of this work lies in the combination of the Digital Twins concept and data-driven models to capture the spatio-temporal features of a zone of interest and to facilitate efficient HVAC system controller through WSN.

II. The System Model

The system model streamlines the intricate components, enabling seamless data collection, transmission, and processing. It supports the analysis and control sub-modules, which initiate controlling signals according to predefined strategies, aiming to optimize the HVAC system control.

A. The Proposed Digital Twin Approach

The digital twin stands as a crucial and advancing technology in the realm of digital transformation and intelligent enhancement. Leveraging data and modeling, the digital twin has the ability to perform tasks such as monitoring, simulating, predicting, and optimizing [17]. As shown in Fig 1, the

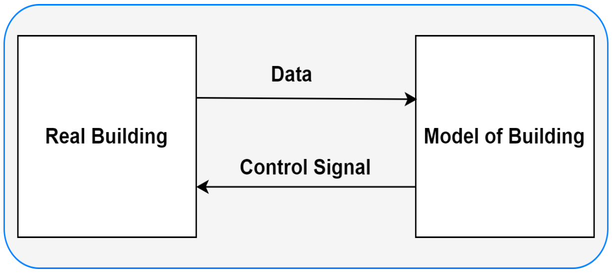

Fig. 1. The Proposed Approach to optimize HVAC System

propsed system model is composed of three main structures, namely: Physical Twin (PT) represented by the real building, the communication interface, which exchanges data and control signals in regular and industrial communication protocols. In addition, the Digital Twin (DT) is represented by the building model. The latter is proposed to incorporate two layers namely cloud and edge. Devices on the edge layer provide cross-communication between the PT and the cloud, in addition to perform online data analysis and controlling functionalities. While, the cloud layer is incorporated to empower the solution by enabling large-scale data collection and storage, which may be used for efficient model training and validation tasks. The following sections describe in more detail the modules and functionalities of both PT and DT.

B. The Physical Twin of The Candidate Building

The term 'physical twin' is a domain-specific concept that refers to an entity of interest characterized by its tangible existence [17]. In the presented case, the physical twin (PT) is represented by the actual candidate building with several sensors and actuators, in addition to the AC and heating modules. The main objective is to enable efficient utilization of the later modules that balance energy consumption with a sufficient level of indoor comfort. Sensor nodes are deployed to collect observations that represent different parameters contributing in forming the environment. Information about the environment around the building plant can be resourceful. Temperature, humidity, power consumption, and conveyor vibration are valuable information for maintaining efficient HVAC system functionalities. On the other hand, actuators are deployed to apply any action as a result of the processing of the raw sensory data performed by the DT counterpart.

As shown in Fig2 the physical twin consists of the zone of availability, sensor nodes and the AC and heating system modules and their control panel of the candidate building. The first corresponds to the location where sensors are deployed, and it is the environmental conditions that are controlled by the HVAC system. In this study a residential building in Montreal-Canada is used where several sensors are deployed to collect data mainly temperature and humidity, in addition to outdoor temperature obtained from nearby weather station. On the other hand, the candidate building is equipped with two separate heating and ventilation systems. Currently, these systems are operated using simple On/Off controllers, leading to excessive energy consumption. The initial goal of our proposed solution is to reduce energy costs by implementing an AI-based controller. This process entails gathering data from the candidate building to create a model that accurately reflects the system's behavior. These system dynamics can exhibit both linear and nonlinear characteristics, and understanding them is crucial to enhance the performance of any proposed controller, particularly Model Predictive Control (MPC)-based controllers.

Lastly, a communication interface is presented to bridge communication between senors nodes at the PT and the edge layer at the counterpart DT. This interface supports various communication protocols, including WiFi, Zigbee, Bluetooth, and LoRa. In our implementation, sensors transmit their data to an intermediary gateway using Bluetooth technology. This gateway is an integral part of the Physical Twin (PT) and is connected to the building network via WiFi. However, due to its resource limitations, the gateway is incapable of processing data or carrying out decision-making tasks. Consequently, it forwards the raw sensory data over WiFi, in our specific case, to a more capable gateway device, such as a Raspberry Pi, which is a crucial component of the the DT.

III. The Proposed Edge of the CB Digital Twin

Generally, the DT of the candidate building is designed to encompass the virtual components necessary for processing the data collected by the Physical Twin (PT). More specifically, the raw sensory data transferred from the PT undergo a series of transformations. These steps are essential to convert the data into meaningful information that can be effectively utilized for decision-making tasks, as exemplified in this study. In this study, the DT comprises primarily two layers: the Edge layer and the Cloud layer.

Real-time analysis of raw sensory data is imperative due to its pivotal role in promptly detecting patterns and anomalies, thus positively influencing event outcomes. To address this necessity, the edge layer of the DT is strategically designed to facilitate the deployment of Machine Learning (ML) models at the edge and in close proximity to data sources. Several submodules are incorporated within the edge layer to enable data preprocessing and to handle the prediction tasks, as well as, to keep tracking of the model performance. The later, which in our case is LSTM, serves as an input source for the MPC controller component that carries out the implementation of predefined controlling strategies intended to ensure efficient HVAC system functionalities. The following subsections describe the modules entailed in the edge layer of our DT.

A. Data Acquisition

In the proposed architecture, data is initially collected through data acquisition APIs.These APIs act as an intermediary, bridging interactions between sensor nodes and edge devices by providing a seamless abstraction layer that supports

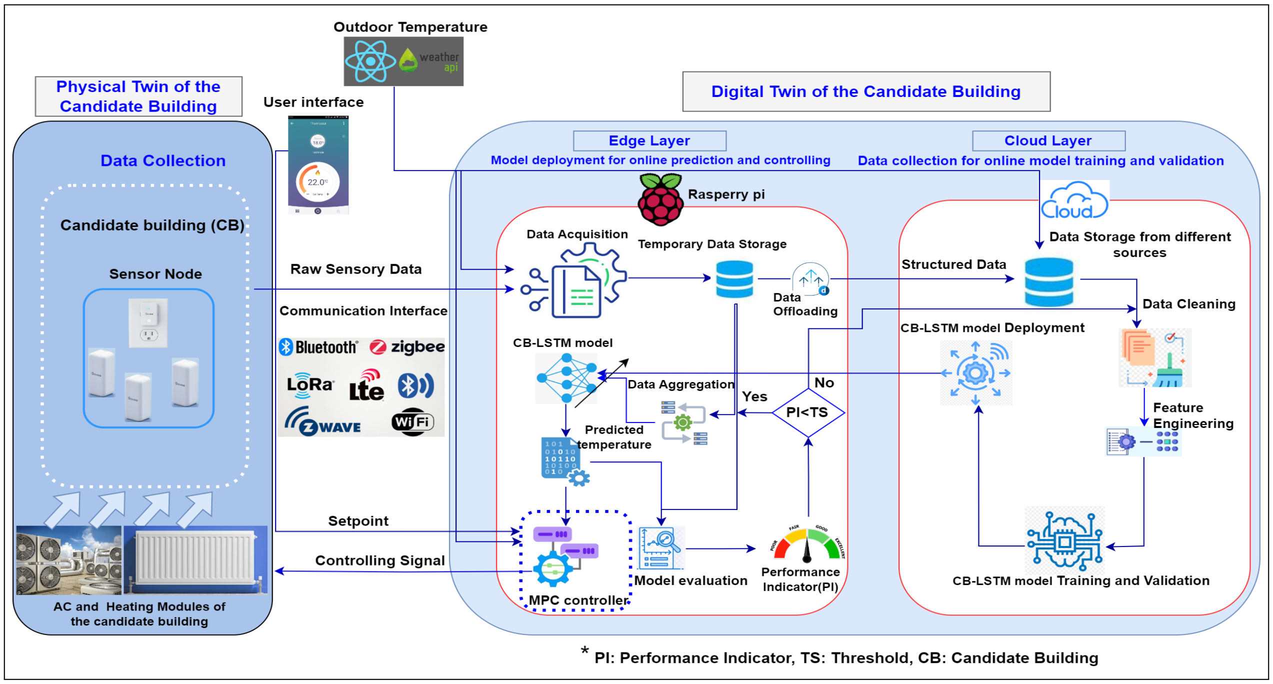

Fig. 2. The Proposed Implementation of the digital twin of the candidate building

bi-directional communication. Several network protocols, as depicted in Fig.1, including Zigbee, WiFi, and LoRA, play a crucial role in establishing effective network level communication. These protocols have been designed to operate effectively in diverse environments, catering to a wide range of use cases by enabling communication over varying distances [11] [12]. On the other hand, data acquisition tasks can be efficiently accomplished by means of application protocols such as MQTT (https://mqtt.org/), and CoAP [13].

Typically, raw sensory data is transmitted to the gateway device in the form of time series sequences that require the use of specialized time series databases (TSDBs) such as Redis (https://redis.io/), and InfluxDB (https://www.influxdata.com/). The edge layer storage unit is temporary, and data will be transformed in later steps to the aggregation module to formalize it for online prediction tasks. Additionally, the data will play a role in assessing the model's performance. Finally, collected data will be periodically transferred to the larger and permanent data storage in the cloud to empower model training and validation tasks.

B. Data Aggregation

This sub-module is designed to convert retrieved data packets into a format compatible with the deployed CB-LSTM model. It encompasses various feature engineering steps followed , including the extraction of new features from received data. More specifically, extra features, such as day/night, and week days or weekend, can be extracted, from the timestamp. Additionally, the data will be shaped to ensure consistency with the data format used during the model training and validation phases.

C. CB-LSTM Model

This module is designed to carry out the execution of the deployed model, enabling it to perform real-time prediction and classification tasks. It has the flexibility to accommodate single or multiple models to meet the edge layer's service requirements. In this study, we employ an LSTM model specific to the candidate building. Recent data is continually and periodically transmitted from the data aggregator module in vectors of a predefined time window (TW) size.

As discussed in [14], the recursive forecasting strategy is a widely adopted approach for predicting several future steps that cover a predefined prediction horizon. In this approach, collected sensory data are framed with fixed-length queue, aligned with a chosen time window. The proposed model forecasts one step into the future at each iteration within the prediction horizon. During each prediction step, the most recent forecasted value is incorporated into the sequence, and the oldest value is removed. The recursive strategy is favored for edge computing due to its lower computational resource requirements compared to the direct strategy.

D. MPC controller

Model Predictive Control (MPC) operates by employing a predictive mathematical model of the building's dynamics and continually conducting online optimizations to establish a control strategy that attains the desired indoor conditions while minimizing a specific parameter, such as energy consumption [15]. Thus, values predicted by the LSTM model will be transferred at a later step to be used as inputs for the MPC controller. The MPC controller utilizes these received predictions, in conjunction with other parameters such as outdoor temperature retrieved online via an API and the setpoint received from a user interface, to calculate the necessary values for optimizing HVAC system functions. These calculated values are subsequently transferred for application to the PT.

E. Model Evaluation

For our models deployed on edge devices to facilitate real-time predictions, we've introduced a model monitoring and evaluation module. This module plays a central role in our approach by tracking model performance and ensuring that the current ML model outcomes are aligned with the actual observations.

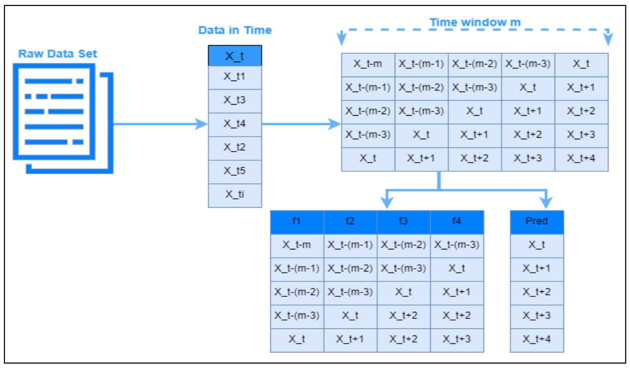

Fig. 3. Time Series data windowing step

The Model Performance Indicator (PI) is iteratively computed based on the availability of the actual data corresponding to previously predicted data within a specific prediction horizon. This process can be configured to run periodically (e.g., hourly or daily) but is likely to be more effective when monitored in shorter intervals. This can be achieved by setting a model performance threshold and monitoring when the model's performance falls below the predefined threshold. Consequently, empowering triggering retraining procedures in real time to maintain model credibility.

IV. The Proposed Cloud of the CB Digital Twin

This section presents an in-depth explanation of the modules designed for initializing the CB-LSTM model, which is planned for deployment and execution in subsequent stages at the edge layer. It outlines the steps to construct ML models, resulting in models for deployment and online predictions at the edge layer. Within this framework, basically, two tasks are intended to be carried out, as elaborated below:

1) Model Creation: Initially, this task represents the model life cycle, commencing with the creation of the training data set. This data set may originate from one or multiple data sources, depending on the service's requirements to be equipped with the trained model.
2) Model Retraining: This task is triggered by receiving a retraining request from an edge device or through periodic configuration. Its purpose is to create a new model by updating the data set with new instances and then comparing the accuracy of the two models.

The following briefly highlights the common steps to promote acceptable model performance.

A. Data cleaning

This step involves addressing missing data points, which have the potential to negatively impact the performance of our model. Since we are working with multiple inputs, each associated with its respective time stamps, it becomes crucial to ensure that all data points are aligned in terms of their time stamps. To elaborate further, certain issues in the data collection process, often stemming from operational factors related to sensors, can result in time stamp incompatibility between sensory data and data collected from other sources.

B. Feature Engineering

This step involves data transformation for time series prediction tasks. Time series data is typically in the form of sequential observations over time, which may not be directly suitable for many machine learning algorithms that expect tabular data as input. Hence, a windowing or sequencing step is needed to divide the time series data into fixed-length windows or sequences as depicted in Fig 3. Each window represents a specific time interval and contains a subset of the time series observations. This step helps capture temporal patterns and dependencies in the data.

C. Model Evaluation

For model validation we applied the mean square error (MSE) and R-squared (R^2) as given in equations (1) (2).

$$MSE = \frac{\Sigma_{i=1}^{n}\left(T_i' - T_i\right)^2}{n} \tag{1}$$

Where T_i' is the predicted temperature, while T_i is the actual temperature , and 'n' denotes the total number of observations used for analysis.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\Sigma i\left(\hat{T}_i - \bar{T}_i\right)^2}{\Sigma i\left(T_i - \bar{T}_i\right)^2} \tag{2}$$

Where T_i is the real temperature,$\hat{T}_i$ is the predicted temperature and $\bar{T}_i$ is the mean of all the temperature values. Mainly, the R^2 formula computes two parameters: the residual sum of squared errors of a regression model, denoted as ***(SSres)***, and the total sum of squared errors, denoted as ***(SStot)***.

D. LSTM model

The Long Short-Term Memory (LSTM) model is a specialized type of recurrent neural network (RNN) that excels at capturing and modeling sequential data with dependencies over extended time intervals. Unlike traditional RNNs, LSTMs are equipped with gating mechanisms that allow them to selectively remember or forget information from previous time steps, making them highly effective in handling vanishing gradient problems and retaining crucial context information. This unique architecture makes LSTMs particularly well-suited for a wide range of tasks, including time series forecasting, natural language processing, speech recognition, and various other sequential data analysis applications [16].

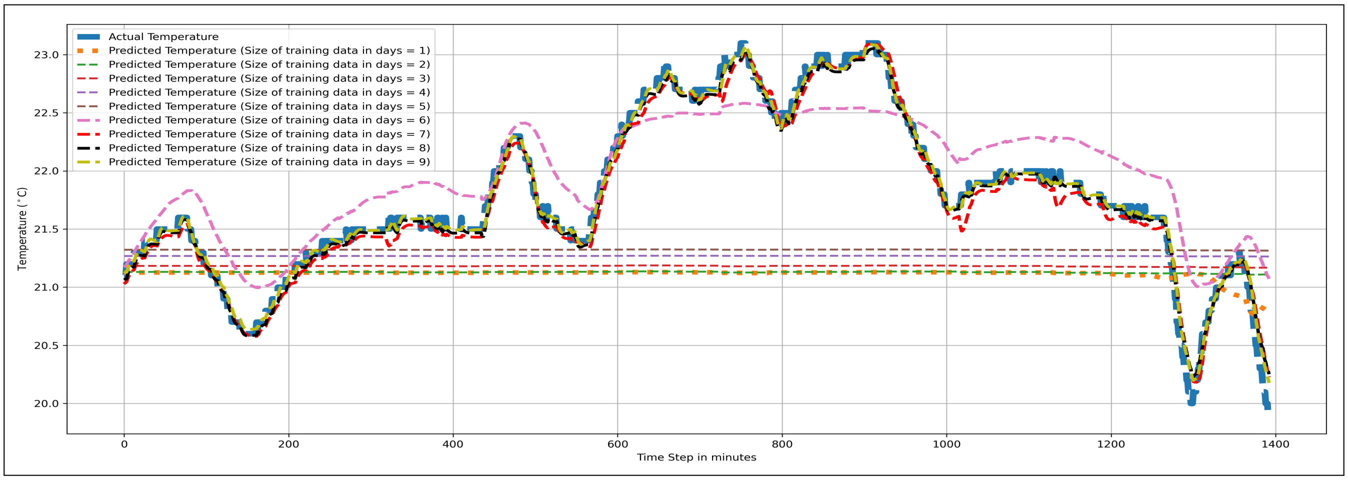

Fig. 4. The LSTM model Performance inline with the size of data in days

V. Experimentations and Results

This section presents the settings applied to gain the results, as well as, present and discuss the obtained results.

A. Data Collection

This process involved the deployment of several sensors and a gateway device in the CB located in Montreal area. More specifically, two zones of the CB have been investigated namely the community and meeting rooms. Set of Govee[1] sensors are used to collect the indoor temperature and humidity, while the outdoor temperature was collected from nearby weather station. This data collection took place over a period extending from August 11, 2023, to August 31, 2023.

- Indoor temperature and humidity readings were recorded at one-minute intervals, resulting in a total of 29,253 samples.
- Outdoor temperature data was captured at one-hour intervals, resulting in a total of 504 samples.

B. Data Preprocessing

Indeed, dealing with the incompatibility in the number of samples required additional data preprocessing efforts to achieve a consistent data collection granularity of one minute. This alignment of data granularity was essential to ensure that the data from both indoor and outdoor sources could be effectively used together for analysis and modeling.

Furthermore, feature engineering steps were applied. Firstly, to transform data samples to sequences that can be divided in later steps to model inputs and outputs. Consequently, in the present implementation a time window of 30 minutes has been selected and our data has been structured accordingly. Secondly, several features have been extracted from the time stamp including day/night, season and weekdays/weekend. These features are categorial and they were encoded to take either 1,0 values. Thus, a normalization process also carried out to avoid any biased model behavior due to different feature scales.

[1]https://ca.govee.com/collections/smart-sensors

C. Model Creation and Validation

The approach adopted for model development and evaluation closely simulates real-time conditions. Initially, the dataset was divided into two distinct segments: one for training and validation and another for testing the model's performance. Specifically, given our dataset spanning 21 days of data, the most recent 2 days' data were reserved for testing, while the remaining 19 days were allocated to training and validation. The training and testing process operates iteratively, beginning with the use of data from the last day in the training dataset and progressively moving backward to the first day (19 to 1). For each day in this sequence, the data from that day was split into an 80% portion for model training and a 20 % portion for validation. Subsequently, the model was applied to the testing dataset, and performance metrics, including accuracy measurements, were recorded. Additionally, the execution time of the model during each iteration was logged. This iterative process continued, with the dataset progressively accumulating data from each subsequent day, and the aforementioned steps were repeated accordingly.

We configured the LSTM model with 50 neurons, a single dense layer, and applied the mean absolute error loss function in conjunction with the 'adam' optimizer algorithm. Additionally, the model was trained and tested following a multi-input and one-output strategy.

Python 3.11 with Panda, Numpy and Matplotlib packages are used for the data preprocessing and results visualization, while the Keras APi is used for model design, creation and validation.

D. RESULTS and DISCUSSION

The first experiment illustrates the duration of data accumulation required on a daily basis to maintain satisfactory model performance. This information enables us to offer users an estimate of the time it will take to fully prepare the solution, as presented in Fig 2, for operational use. As depicted in Fig 4, our model achieved stable and acceptable performance after 9 days of data accumulation. More specifically, Fig 4 compares model performance across training and validation iterations.

Each iteration can be perceived as a model that trained and validated using a different dataset size, while using a uniform testing dataset for all iterations.The testing data set is plotted on the X-axis of the Fig 4 and it represents 1 day of data that was not involved in the training and validation phase.

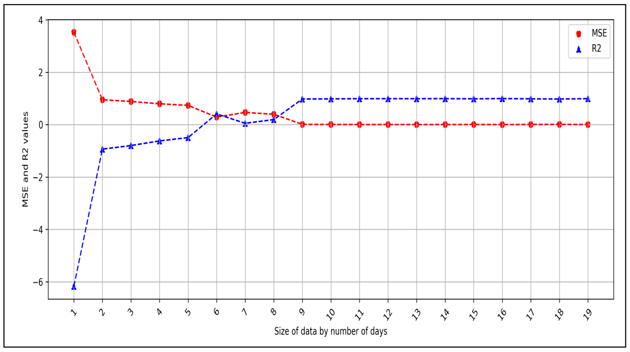

Fig. 5. The LSTM model Performance measurements in accordance to the size of the used data

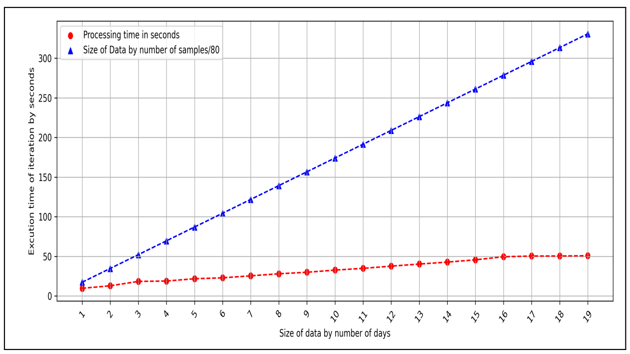

Fig. 6. Execution time by iteration and size of used data

A similar conclusion drawn by Fig 5 that shows the variations of the model performance indicators used in this work namely MSE and R^2. Both accuracy measures shown started poorly but with the accumulation of the dataset and starting from day 9 and onwards the performance has gotten stabilized.

However, the presented results can be further enhanced with better performance achieved in fewer days of data by fine-tuning certain training parameters, such as the number of epochs and batch size. Nevertheless, it is important to set a balance between improved results and efficient execution time, which is crucial, especially when retraining and deploying the model may be necessary as presented int the system model shown in In Fig 2. In Fig 6, execution time is illustrated in relation to data size. To accommodate presentation, the Y-axis data size is scaled down by a factor of 80.

VI. Conclusion

In this work, we have introduced a data-driven and WSN-based digital twin system model for efficient HVAC systems. Mainly, we gave a detailed description of the proposed architecture revealing the potential of such a combination. In addition, a candidate building (CB)-LSTM model was developed using a dataset collected from a candidate building. The steps followed to develop and evaluate the CB-LSTM model has been detailed showing the impact of the period needed to collect the data from a real building in order to obtain reliable model performance. In addition, the correlation between the size of data needed and the time needed to create and validate the model along with the performance is considered in this work. Future research will be directed into two streams. First, increasing the model's granularity by considering more relative environment parameters. Second, investigating possible mechanisms and factors for efficient model deployment and real-time predictions on gateway devices.

Acknowledgment

This work was supported by the MITACS Accelerate program and with the collaboration of Quantolio Technologies Inc.

References

[1] Abdul Afram et al. "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system." Energy and Buildings 141 (2017): 96-113.
[2] https://madeinca.ca/hvac-statistics-canada/
[3] Bhaskoro, Petrus Tri, Syed Ihtsham Ul Haq Gilani, and Mohd Shiraz Aris. "Simulation of energy saving potential of a centralized HVAC system in an academic building using adaptive cooling technique." Energy Conversion and Management 75 (2013): 617-628.
[4] Budaiwi, Ismail, and Adel Abdou. "HVAC system operational strategies for reduced energy consumption in buildings with intermittent occupancy: The case of mosques." Energy conversion and management 73 (2013): 37-50.
[5] Duffy, Matthew J., Marion Hiller, David E. Bradley, Werner Keilholz, and Jeff W. Thornton. "TRNSYS–features and functionality for building simulation 2009 conference." (2009).
[6] Zhao, Jie. "EnergyPlus model-based predictive control (EPMPC) by using MATLAB/SIMULINK and MLE+." (2013).
[7] Zhou, Datong P., et al. "Quantitative comparison of data-driven and physics-based models for commercial building HVAC systems." In 2017 American control conference (ACC), pp. 2900-2906. IEEE, 2017.
[8] Abida, A., and P. Richter. "HVAC control in buildings using neural network." Journal of Building Engineering 65 (2023): 105558.
[9] Ra, Seon Jung, Jin-Hong Kim, and Cheol Soo Park. "Real-time model predictive cooling control for an HVAC system in a factory building." Energy and Buildings 285 (2023): 112860.
[10] Minerva, Roberto, Gyu Myoung Lee, and Noel Crespi. "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models." Proceedings of the IEEE 108, no. 10 (2020): 1785-1824.
[11] Anani, Wafaa, Abdelkader Ouda, and Ali Hamou. "A survey of wireless communications for IoT echo-systems." In 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), pp. 1-6. IEEE, 2019.
[12] Abdullah Ahmed Bahashwanet al.,. "Review on common IoT communication technologies for both long-range network (LPWAN) and short-range network." In Advances on Smart and Soft Computing: Proceedings of ICACIn 2020, pp. 341-353. Springer Singapore, 2021.
[13] Almheiri, Alyaziya, and Zakaria Maamar. "IoT Protocols–MQTT versus CoAP." In Proceedings of the 4th International Conference on Networking, Information Systems & Security, pp. 1-5. 2021.
[14] Taieb, Souhaib Ben, and Gianluca Bontempi. "Recursive multi-step time series forecasting by perturbing data." In 2011 IEEE 11th International Conference on Data Mining, pp. 695-704. IEEE, 2011.
[15] Rockett, Peter, and Elizabeth Abigail Hathway. "Model-predictive control for non-domestic buildings: a critical review and prospects." Building Research & Information 45, no. 5 (2017): 556-571.
[16] Lindemann, Benjamin, Timo Müller, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. "A survey on long short-term memory networks for time series prediction." Procedia CIRP 99 (2021): 650-655.
[17] Tao, Fei, Bin Xiao, Qinglin Qi, Jiangfeng Cheng, and Ping Ji. "Digital twin modeling." Journal of Manufacturing Systems 64 (2022): 372-389.