A Reconfigurable Strategy for Internet-of-Things for Smart Buildings

Xichun Yang[1], Ananda Maiti[1], Alexander Kist[2]

[1]University of Tasmania, Newnham, Australia
[2]University of Southern Queensland, Toowoomba, Australia
anandamaiti@live.com

Abstract. The Internet of Things (IoT) is being used for various applications where data is collected over time. The data generated in time series is then analyzed to establish patterns, identify problems, and make different decisions. The amount and quality of data generated this way depend on the IoT system's configuration. If the configurations are left static, the IoT systems can consume excess resources, e.g., battery or network infrastructure, to produce a quantity of data or not produce quality data. This paper proposes an approach to reconfigure IoT systems dynamically to consume relatively lower resources while still generating acceptable quality of data fit for an application. Configuration can include several parameters, e.g., frequency of data collection, spatial orientation, and authentication. We first propose a generic reconfiguration framework and then mainly discuss the time aspect of reconfiguration in this paper. In this approach, the frequency of data collection is dynamically altered depending on the current data being generated and the predicted future data points. We show that given an absolute maximum frequency where the quality of the output data set is the best and a minimum frequency for an IoT device in a system where the resource consumption is lowest, the dynamic reconfigurable IoT approach can operate with optimized time gaps, still producing good quality data. The proposed approach uses a Smart Building IoT system as a case study.

Keywords: Internet of Things, Smart Buildings, Anomaly Detection, Predictive Analysis.

1 Introduction

Smart buildings have become popular worldwide to increase users' performance efficiency and operational options. Such environments generate a large data volume that almost doubled between 2010 and 2020 [1]. It has been forecasted that the smart building market will rapidly grow from 2020 to 2025 with a 23% annual growth rate [2]. The investments in smart buildings aim to reduce energy consumption, improve management efficiency, and comfort people's lives [3]. The smart buildings revolution currently has three main objectives. The first is the reduction of energy consumption. Secondly, the detection capability of sensors could improve the management efficiency of smart buildings, such as event tracking and warnings. The third driving force is to provide comfortable and convenient lifestyles to users with an integrated IoT system. This paper focuses on the first and second aspects of the IoT system. A large amount of data is generated from multiple devices over time. It is time and energy-intensive to either reduce the big data to a more minor data set or process it whole. Also, radio communications can consume battery power reducing their working life. There is an increasing drive to deregulate the cloud based IoT architecture by pushing a reasonable amount of computing to the edge of IoT in the form of edge computing. Such critical decisions can be made locally at an individual device or with a cluster of devices.

In general, smart buildings rely on a network of sensors and actuators as part of the IoT systems [4]. Different technical layers in IoT architecture enable flexible deployment of devices, and such "dedicated" things can share data and ultimately provide intelligence, autonomous action, and business value. Reliable, good quality, and timely gathered data sets are vital to smart building IoT systems. Properly deployed and operated sensors and actuators are the foundation of the data collection. These components must be as autonomous as possible with low power consumption characteristics.

The sensor supplies data to the IoT system of smart buildings' time series, which can be the foundation for machine intelligence. This paper's key contribution is to propose a versatile event/anomaly detection system to achieve the efficient reconfiguration of IoT systems in smart buildings. Efficiency in the current context is the ability to detect anomalies with fewer data. This way IoT system can be reconfigured to generate just the right amount of data required to capture the relevant events or anomalies. Several techniques are used to realize the proposed reconfiguration. This includes a time series prediction model to capture the characteristics of the time-series data. Dynamic Time Warping (DTW) combined with an unsupervised anomaly detection technique and isolation forest is used to find similarities in detected anomalies.

This paper first reviews the related work on smart buildings, IoT, prediction, and anomaly detection mechanisms in Section 2. Section 3 discusses the paradigms of the proposed reconfigurable IoT. A time series based reconfigurable IoT system is shown in section 4 and Section 5.

2 Related Work

2.1 Smart Buildings

A smart building uses big data, IoT, AI, and other scientific technologies to help people be more comfortable working and studying environments. It also improves the production efficiency of the target environments. Smart building research covers smart homes, factories, and agriculture [5-9].

The development and optimization of smart buildings rely on a sensor network, which is now part of more complex IoT systems. Firstly, IoT helps create efficient energy management solutions for smart buildings [10, 11]. The sensors in the smart building can monitor the various environmental factors, e.g., temperature, humidity, and personnel activities in the building, in real-time. Secondly, many facilities in the smart building use preventive maintenance to ensure the proper operation of equipment.

Through time series forecasting, machine learning, and other methods, initial problems could be predicted, and the smart building could reconfigure the IoT system for the proper operation of types of equipment. Another significant feature the IoT system brings to smart buildings is access to real-time data [12-15]. Building managers could develop and improve management strategies by visualizing and analyzing collected data. Sensors installed in a smart building could monitor the corresponding data in real-time and report problems once they are found, including abnormal temperatures and fire alarms and PIR sensors monitoring the real-time occupancy in the smart building. The collected data can also optimize the space configuration and maximize the use of reasonable space, which can optimize the efficiency of the building.

The technologies used to implement an IoT system in smart buildings generate data in multiple formats, including time series, identification/authorization, and metadata [16-19]. In this paper, we consider time series data from continuous environment monitoring to propose a reconfigurable IoT system. However, the reconfiguration can impact all other kinds of data as well. While the importance of IoT technology for smart buildings is beyond doubt, collecting meaningful data can be improved. By analyzing and processing the sensor's data, the smart building's efficiency can be maintained while reducing energy consumption and device maintenance.

2.2 Prediction Mechanisms

Time series data is often used to predict or forecast the future. Time series forecasting predicts time series data, a field of machine learning [20, 21]. Time series is a set of data points collected at different time intervals. Time series forecasting figures out the regular patterns in the past time series data and predicts future events.

Time series forecasting analysis has various models, and five of them will be compared in this research, including Autoregression (AR), Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), Simple Exponential Smoothing (SES), Vector Autoregression (VAR). The AR model creates a regression equation through the correlation between historical values, which can achieve the purpose of future prediction. This process becomes an autoregressive process. Similar to the AR model, ARMA is an effective tool for analyzing stationary time series. In the ARMA model, the purpose of the autoregressive process is to detect the logic of data changes and alterations. The moving average process can handle the problems caused by randomness, and the ARMA model has more advantages in applications and is more extensive.

2.3 Anomaly Detection Mechanisms

In the smart building IoT system reconfiguration, the detected anomaly is the leading cause of it. Anomaly detection is used to identify events that are not expected. These events play significant roles in IoT system reconfiguration. Taking the reconfiguration of temperature sensors in the IoT system as an example, timely anomaly detection of indoor and outdoor environments is the core of smart building management [22]. The anomaly detection technique has been applied in different aspects of real life, including bank fraud, medical service, network intrusion detection, traffic safety, and industrial equipment failures.

In terms of the anomaly detection algorithm, based on the amount of labeled data, anomaly detection could be separated into three categories, supervised, semi-supervised, and unsupervised anomaly detection models [23-27]. This section mainly presents the isolation forest algorithm. Isolation Forest is based on integrated learning to achieve outlier detection, which has almost become the first choice of outlier detection algorithms [28].

2.4 Anomaly Similarity

In the field of time series study, comparing the similarity of the different time series is essential. The traditional Euclidean distance algorithm compares the points of two different time series with the same length through a one-to-one comparison [29]. The main feature of Euclidean distance is that it requires time series to be aligned. Although the time series are similar, the Euclidean distance may be too large because the phase difference between the two series and the peaks are not aligned. The application of Euclidean distance in time series similarity measurement has its limitations. For example, in the field of speech recognition, speaking speed is different in various situations. In the current context, the Euclidean distance becomes invalid because the length of the time series is inconsistent. Dynamic Time Warping (DTW) is proposed as a solution to overcome the shortcomings of Euclidean distance. Compared with Euclidean distance, DTW compares two time series in one-to-many mode [30-32]. Time warping means that a point in a time sequence can correspond to multiple consecutive points in another time series.

This research uses DTW with two time series, each from a different data collection strategy with a different time gap. It is used to determine the level of accuracy in identifying the anomalies in each period.

3 Paradigms of the Proposed Reconfigurable IoT

3.1 Predictions and Anomaly

Predictions and anomalies are the critical parts of the proposed reconfigurable IoT. In the smart building IoT system, a significant amount of time series data is stored in the knowledge base for real-time analysis and training purposes. Real-time data, sensor, and trained data complete predictive analysis and anomaly identification with machine learning. The first step is to predict the next data point based on the existing training data. Then we calculate the anomalies with isolation forest based on the difference between the predictions and the actual sensor values. Thus, we have a series of predicted and actual values for each time point in the time series. Due to the difference between prediction and actual data, anomalies can be detected.

3.2 Phases of Anomaly Events

A sensor collects data at regular intervals with a time gap (τ).

$$D = \{d_0,\ d_1, \dots\ d_t\} \qquad \dots (1)$$

For each incoming data at time t, a predicted value p_t may be calculated based on a subset of D. As such, we have a set of predicted sensor values,

$$P = \{p_0,\ p_1, \dots\ p_t\} \qquad \dots (2)$$

where, p_i is predicted with $\{p_{t-\alpha} \dots p_t\}$. An anomaly event is defined as a continuous sequence of k specific sensor values at a given time t based on the difference between P and D.

$$e_t \subset D \qquad \dots (3)$$

Multiple sensors may be dynamically associated with an event concerning time as well, and a sensor may also be shared among multiple events. In the simplest form, a sensor's values going beyond a given range can be considered an anomaly event associated with the sensor(s).

3.3 Static configuration vs. Dynamic configuration

IoT data is often simple time-series data, or the data is almost always timestamped, allowing the developers to analyze the data in time. The *time gap* (τ) between the data points in the time series can be static or variable. *Static reconfiguration* means the data is collected with a fixed time gap. A large *time gap* results in sparse data points but saves communication or storage-related energy. A smaller time gap means a higher resolution of data but results in a big data set.

Compared with the static state, the dynamic state is multipurpose. Dynamic reconfiguration can mean a change of orientation, data collection frequency, sensor groups, and locations, or a combination of any of these. In terms of event-triggered reconfiguration, the system constantly watches for a specific event with a dedicated sensor. However, this may not be known before and hence is hard to implement. Possible events can be detected by time series analysis, and dynamic reconfiguring can allow the IoT system to focus on finding and capturing a specific event.

In this paper, we focus only on the time aspects (τ) for detecting and reconfiguring the IoT system in terms of data collection frequency.

4 Implementation and Results

4.1 Data and Simulation Environment

Any sensor can generate numeric inputs, e.g., temperature, humidity, illumination, gas levels, and smoke. In this research, we used a temperature sensor. However, the sensor's construct and structure are irrelevant to the proposed framework for detecting events or anomalies, as any sensor can generate a time series. Each row of the raw sensor dataset consists of a

{timestamp, sensor value}

In the simulation, we see the effect of altering the data collection strategy by skipping some of the data for prediction and anomaly detection.

Figure 1 depicts the operation of the reconfigurable IoT. The stream of data is received from IoT devices. Then a series of steps are followed for each data arrival or the end of the time gap:

Step 1. Based on α data points, p_t is determined and put in the predicted values set. Initialize a buffer time γ.

Step 2. Next, the isolation forest is used to determine the anomalies between the actual and predicted values.

Step 3. Then DTW is used to find similar matching anomalies within those anomalies from Step 2.

Step 4. If the anomalies increase in number, the time gap is reduced by 1 minute.

Step 5. If the number of anomalies decreases and the buffer time of γ has passed without any more increase in the number of anomalies, the time gap τ increases by 1.

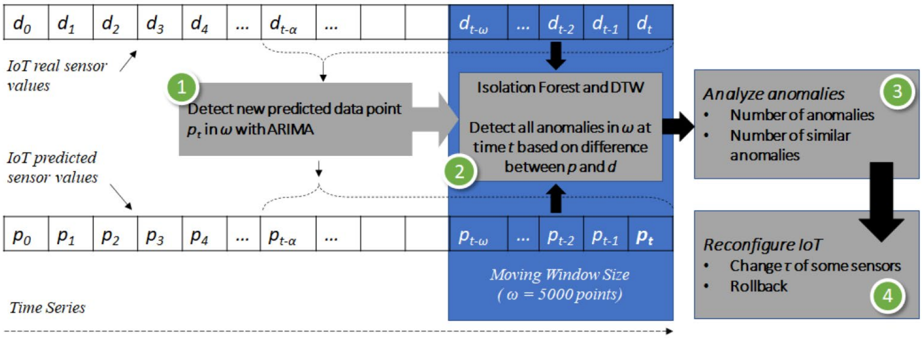
d_0 d_1 d_2 d_3 d_4 … $d_{t-\alpha}$ …
$d_{t-\omega}$ … d_{t-2} d_{t-1} d_t
IoT real sensor values
1
Detect new predicted data point p_t in ω with ARIMA
Isolation Forest and DTW
Detect all anomalies in ω at time t based on difference between p and d
2
Analyze anomalies
• Number of anomalies
• Number of similar anomalies
3
IoT predicted sensor values
p_0 p_1 p_2 p_3 p_4 … $p_{t-\alpha}$ …
$p_{t-\omega}$ … p_{t-2} p_{t-1} p_t
Moving Window Size (ω = 5000 points)
Reconfigure IoT
• Change τ of some sensors
• Rollback
4
Time Series

Fig. 1. The reconfigurable IoT implementation diagram

The system maintains a list of all anomalies θ_ω within a time period ω. To prove the effectiveness of the proposed system, we compare the reconfigurable IoT with the dynamic and static configurations in terms of real-time properties and data quality/quantity. The following are the measure for comparison:

To measure the system's efficiency, we term "Data generation/usage" as the amount of data that would have been needed if this was implemented in real. For example, for a time gap of 1 minute, every row in the data is used, i.e., the data generation/usage is 100%. If every 5th row is picked, then it is 20%. If every 15th row is picked, then it is 6.6%. It essentially indicates the amount of data used from the dataset to achieve the reconfiguration results. Actual implementation will generate the data at the same time intervals as the simulations.

The sensor is merely observing the environment and the prediction of an anomaly only leads to a decision on whether to increase or decrease the data collection rate.

The IoT reconfiguration system has the following aims:

- Reduce the data generated/used by each device, and the system could reduce cost, power consumption, radio traffic, and storage. This can be determined by the amount of data used/generated and the number of anomalies detected at the corresponding times.

- Along with reducing data generation and consumption, the quality of the data collected must also be maintained. This means the data collected must be usable for future offline analysis.
- Achieve real-time processing, i.e., perform well in detecting anomalies and collecting relevant information, as if the system is running at the lowest possible value of time gaps. This means the anomalies must be detected in dynamic configurations at approximately the same time as a static configuration.

For example, considering anomaly detection based on the weather condition data, there would be increasing data points from the sensors with dynamic intervals during abnormal weather conditions. Under normal circumstances, the data points collected by the temperature sensor will be less and at static intervals. An extensive training data size *α* in time series d_t-*α* to d_t can ensure the accuracy of predictions. We use the static parameters for the simulation in this paper, as in Table 1.

If the number of anomalies in the time frame increases, we increase the frequency of data collection in real time (online). A higher resolution of the data collected can help identify repeating anomalies in offline analysis.

Table 1. Simulation Parameters

Parameter	Value
Total data points in data set	13000+
τ_L in the data set (lowest and uniform)	1 minute
Training data size α	5000
Moving Window size ω	5000
Anomaly size σn	5
Prediction Algorithm	ARIMA Model
Initial $\tau = \tau_c$ (*steady* state)	5 minutes

4.2 Simulation Results for different time gap schemes

In this research, the reconfigurable IoT system has been tested in three different time gap schemes in the static configuration. Data collection frequency in the IoT system is set to τ = 5 minutes, 1 minute, and the dynamic configuration with γ = 10 and 5. The results are plotted in Figure 4. With the reduction in the time gap, there are increasing probabilities to find more anomalies and matches in the system. However, for each τ, the number of anomalies follows a similar pattern when they change drastically.

In a dynamic state, the data collection frequency in the system starts every five minutes and changes dynamically according to `ChangeTimeGap`.

Algorithm `ChangeTimeGap`

```
INPUT: $\theta_\omega^t and\, \theta_\omega^{t-1}$
if $|\theta_\omega^t| \geq |\theta_\omega^{t-1}|$ then
    decrease $\tau_{t+1}$ by 1, down to a minimum $\tau_L$ = 1
```

```
else if $|\theta_\omega^t| < |\theta_\omega^{t-1}|$ then
     if $\gamma$ = 0 then
          increase $\tau_{t+1}$ by 1, up to a maximum $\tau_e$ = 5
          reset $\gamma$
     else
           change buffer time $\gamma \leftarrow \gamma - 1$
```

Figure 2 shows that with the parameters stated in Table 1, the dynamic configuration can perform better than τ_S and underperforms compared to τ_L. Figure 3 and Table 2 show the anomaly detection overview. When the time gap changes, it increases the probability of finding more anomalies and similarities within the anomalies. With the increasing number of anomalies, the number of similar anomaly matches increases as well. For example, with τ_L, the number of unique anomalies is 331, the average number of points in anomalies is 13.27, the average time of anomalies is 13.27 minutes, and the average time gap in the anomaly is fixed at 1 minute. The values π_D^5 are closer to τ_L compared to τ_s (which is the *steady* state).

In a static state, the data collection frequency could increase the data usage from 20% (with τ_S) to 100% (with τ_L), but the system operation time is increased by about 2.82s to 7.09s for each τ. The proposed anomaly detection system's dynamic state changes the way IoT observes the environment. With the increasing number of anomalies, the frequency of data collection is increased. When the detected abnormal data stabilizes or reduces, the frequency of data collection is reduced.

The performance of the dynamic state is better than the five minutes and two minutes time gap but could not achieve the outcome of one-minute data collection frequency. The data usage for dynamic reconfiguration is significantly reduced compared to the static time gap of one minute, about 40%.

In the dynamic state, data generation of the anomaly detection system operates around 58.8% of data with 4.17s compared to τ_L. Changing the time gap from five to one dynamically can achieve a high level of anomaly detection but keep the processing or execution time of calculating the anomaly lower than a static configuration.

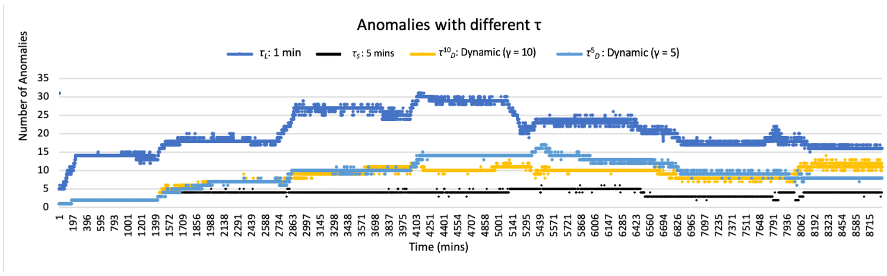

Fig. 2. Real time anomaly detection for different values of τ over a period of over 8000 minutes. Smaller τ generate substantially more anomalies continuously.

4.3 Impact on the matching number of anomalies detected

As mentioned earlier, the proposed system must be proven effective in terms of real-time decisions and longer-term data quality. Figure 3 demonstrates that with the appropriate setting, the dynamic configuration can achieve better results than τ = 5 mins in terms of anomaly detection and better than τ = 1 minute in terms of data consumption. Next, we show that the anomalies detected θ_ω in dynamic τ have similarities to the θ_ω in other static values of τ.

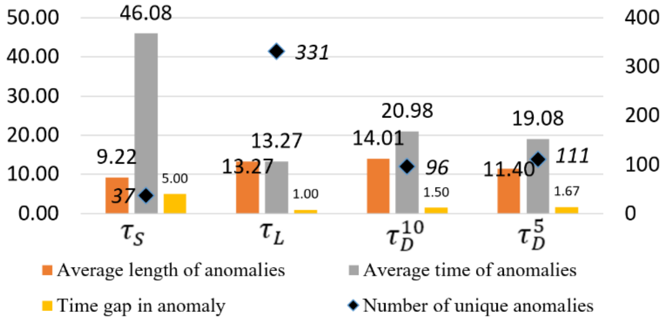

Fig. 3. Anomaly detection overview (number of unique anomalies, average length of anomalies, average time of anomalies, time gap in anomaly)

Table 2: Anomaly Detection Overview

Symbol	Time Gap	Average Execution Time after each τ	Data Generation (compared to τL)
τ_S	5 mins (*steady* state)	2.82s	20%
τ_D^{10}	Dynamic (γ = 10)	4.13s	54.2%
τ_D^5	Dynamic (γ = 5)	4.17s	58.8%
τ_L	1 min	7.09s	100%

This can ascertain that the dynamic configuration can pick up the same anomalies at the same time.

4.4 Prove the real-time worthiness

To check the real-time anomaly detection, the anomalies in θ for several combinations of τ are mapped with DTW. This maps the smaller set of anomalies θ detected with a more extensive τ to the more fragmented larger set of anomalies in a smaller τ in a synchronized manner over time. The analysis process counts the number of times an anomaly in a θ detected with a larger τ configuration matches an anomaly detected with a smaller τ configuration. So, we put the time gaps in the following order according to the number of anomalies they generate (as per Figure 4):

$$\{\tau_S, \quad \tau_D^{10}, \quad \tau_D^{5}, \quad \tau_L\}$$

To describe the real-time worthiness, we define two parameters:

Matching Rate: This is the relationship between the total number of unique anomalies which can be mapped from any τ_i to $\tau_L(\theta(\tau_i, \tau_L))$ to the total number of unique anomalies for τ_i $(\theta(\tau_i))$.

$$MR\ (\tau_i) = \frac{\theta(\tau_i, \tau_L)}{\theta(\tau_i)} \times 100 \qquad (4)$$

For all the possible combinations of mapping τ, the MR is higher than 90%. The results of this comparison are shown in Figure 4(a) when mapping anomalies across multiple τ. The x-axis shows the comparison between the τ_i vs τ_L with $\tau_i > \tau_L$. Each of these mappings is more than 90%, showing that the dynamic configurations can pick up the anomalies almost as good as τ_L. This may be improved by altering the parameters mentioned in Table 1. Figure 4(b) show the same relations but uses the mapping from τ_L to any τ_i, modifying Equation 4 accordingly. Due to fragmentations, the number of anomalies detected at a smaller τ, may not match to a larger τ as the time synchronization does not hold for some fragments.

Note that even if $\tau\tau_{SS}$ appear better in this figure with 97.3% (see Figure 4(a)), it only produces 37 anomalies, 97.3% of which has a corresponding real-time match in τ_L. On the other hand, for $\tau\tau_{DD5}$ the 91.89% matching is from a total of 111 individual anomalies in $\tau\tau_{DD5}$ to 331 individual anomalies with τ_L.

Coverage: This is defined as,

$$Coverage = \frac{\theta(\tau_i)}{\theta(\tau_L)} \times MR\ (\tau_i) \qquad (5)$$

This ratio shows the real effectiveness of the matching between different τ as depicted in Figure 4. The coverage of the τ_S is the lowest and it increases to the highest for τ_D^5.

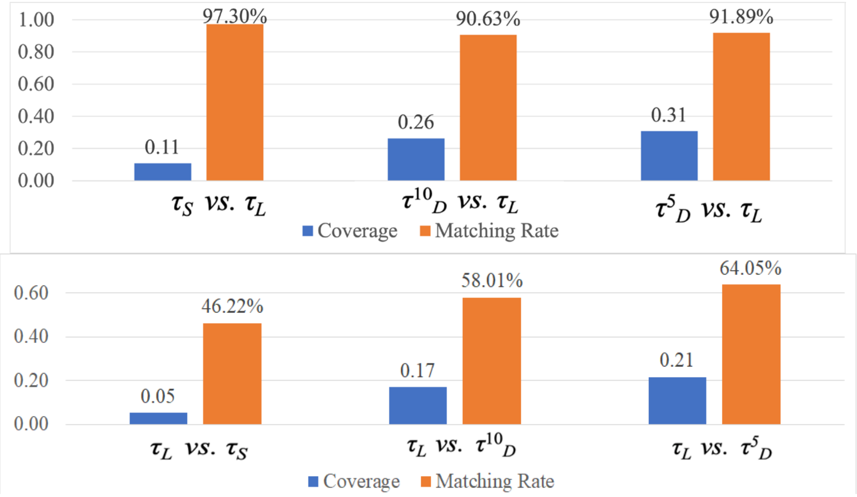

Fig. 4. (a): List of similarities across the different values of τ when time-synchronized with τi vs τL. Most anomalies found in τi has at least one match in τL. (b): List of similarities across the different values of τ when time-synchronized with τL vs τi. The performance, i.e., matching rate increases with lower values of τi

5 Conclusions

Given the large quantity of data generated in various environments, storing a large amount of data is not good. This paper presents a comprehensive analysis of the usability of data collected with multiple time-related contractions. There are four main steps in the IoT system reconfiguration process. The first step is to make predictions on time series data and then apply a reconfiguration plan for the IoT system. Once the reconfiguration is finished, the IoT system is determined whether to roll back by analyzing the feedback. The IoT system reconfiguration of the smart building refers to the change in temperature sensor data collection frequency.

This research can achieve the same level of decision-making accuracy with a lesser amount of data collected to form an IoT environment. Such a method can ensure that once an IoT system is implemented, it can set itself to an optimal strategy to generate data when required, still making the appropriate decisions. This would ultimately reduce network traffic and power consumption.

References

1. Volume of data collected by smart buildings worldwide from 2010 to 2020(in zetabytes). 2016, Statista Research Department.
2. SMART BUILDING MARKET - GROWTH, TRENDS, FORECASTS (2020 - 2025). 2019, Mordor Intelligence
3. Jia, M., et al., Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications. Automation in Construction, 2019. **101**: p. 111-126.
4. Li, L., et al. The applications of wifi-based wireless sensor network in internet of things and smart grid. in 2011 6th IEEE Conference on Industrial Electronics and Applications. 2011. IEEE.
5. Liu, Y., R. Tamura, and Y. Song, Constructing a Smart Home for Future Elders toward All-around Happiness: Taking Connectivity as the Core Element. Applied Sciences, 2020. **10**(16): p. 5690.
6. Kim, J.-A. and J. Jeong, Smart Warehouse Management System Utilizing IoT-based Autonomous Mobile Robot for SME Manufacturing Factory. The Journal of The Institute of Internet, Broadcasting and Communication, 2018. **18**(5): p. 237-244.
7. Cruz, A.C., T. Le, and C. Rogan, Schneider Electric and the Smart Factory. 2019.
8. Lin, J., et al. Blockchain and IoT based food traceability for smart agriculture. in Proceedings of the 3rd International Conference on Crowd Science and Engineering. 2018.
9. Li, N., et al., Smart Agriculture with an Automated IoT-Based Greenhouse System for Local Communities. Advances in Internet of Things, 2019. **9**(02): p. 15.
10. Liang, R., L. Zhao, and P. Wang, Performance Evaluations of LoRa Wireless Communication in Building Environments. Sensors, 2020. **20**(14): p. 3828.
11. Trinh, L., et al. Signal propagation of LoRa technology using for smart building applications. in 2017 IEEE Conference on Antenna Measurements & Applications (CAMA). 2017. IEEE.
12. Lin, Z., X. Liu, and M. Collu, Wind power prediction based on high-frequency SCADA data along with isolation forest and deep learning neural networks. International Journal of Electrical Power & Energy Systems, 2020. **118**: p. 105835.

13. Suma, N., et al., IOT based smart agriculture monitoring system. International Journal on Recent and Innovation Trends in computing and communication, 2017. **5**(2): p. 177-181.
14. Prathibha, S., A. Hongal, and M. Jyothi. IoT based monitoring system in smart agriculture. in 2017 international conference on recent advances in electronics and communication technology (ICRAECT). 2017. IEEE.
15. Djenouri, D., et al., Machine Learning for Smart Building Applications: Review and Taxonomy. ACM Comput. Surv., 2019. **52**(2): p. Article 24.
16. Wei, X., X. Zhang, and L. Geng, Smart Building Temperature Monitoring System using Matplotlib. Journal of Computer Science and Information Technology, 2019. **7**(2).
17. Gao, B., et al. An OAuth2. 0-Based Unified Authentication System for Secure Services in the Smart Campus Environment. in International Conference on Computational Science. 2018. Springer.
18. Balaji, B., et al., Brick: Metadata schema for portable smart building applications. Applied energy, 2018. **226**: p. 1273-1292.
19. Ardagna, C.A., et al., From Trustworthy Data to Trustworthy IoT: A Data Collection Methodology Based on Blockchain. ACM Trans. Cyber-Phys. Syst., 2021. **5**(1): p. Article
20. Hamilton, J.D., Time series analysis. 2020: Princeton university press.
21. Wang, C., et al., A Dependable Time Series Analytic Framework for Cyber-Physical Systems of IoT-based Smart Grid. ACM Trans. Cyber-Phys. Syst., 2018. **3**(1): p. Article 7.
22. Sater, R.A. and A.B. Hamza, A Federated Learning Approach to Anomaly Detection in Smart Buildings. ACM Trans. Internet Things, 2021. **2**(4): p. Article 28.
23. Akcay, S., A. Atapour-Abarghouei, and T.P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. in Asian conference on computer vision. 2018. Springer.
24. Kiran, B.R., D.M. Thomas, and R. Parakkal, An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. Journal of Imaging, 2018. **4**(2): p. 36.
25. Kawachi, Y., Y. Koizumi, and N. Harada. Complementary set variational autoencoder for supervised anomaly detection. in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2018. IEEE.
26. Gimenez, P.-F., et al., RIDS: Radio Intrusion Detection and Diagnosis System for Wireless Communications in Smart Environment. ACM Trans. Cyber-Phys. Syst., 2021. **5**(3).
27. Pang, G., et al., Toward Deep Supervised Anomaly Detection: Reinforcement Learning from Partially Labeled Anomaly Data, in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &amp; Data Mining. 2021, Association for Computing Machinery: Virtual Event, Singapore. p. 1298–1308.
28. Wang, R., et al., Multiple Features and Isolation Forest-Based Fast Anomaly Detector for Hyperspectral Imagery. IEEE Transactions on Geoscience and Remote Sensing, 2020.
29. Maxim, L.G., J.I. Rodriguez, and B. Wang, Euclidean distance degree of the multiview variety. SIAM Journal on Applied Algebra and Geometry, 2020. **4**(1): p. 28-48.
30. Kate, R.J., Using dynamic time warping distances as features for improved time series classification. Data Mining and Knowledge Discovery, 2016. **30**(2): p. 283-312.
31. Hu, J., F. Ma, and S. Wu, Anomaly identification of foundation uplift pressures of gravity dams based on DTW and LOF. Structural Control and Health Monitoring, 2018. **25**(5): p. e2153.
32. Pietroszek, K., P. Pham, and C. Eckhardt, CS-DTW: real-time matching of multivariate spatial input against thousands of templates using compute shader DTW, in Proceedings of the 5th Symposium on Spatial User Interaction. 2017, Association for Computing Machinery: Brighton, United Kingdom. p. 159.