IEC
Institute of Electronics and Computer

Journal of Artificial Intelligence and Systems, 2026, 8, 60-73
https://iecscience.org/journals/AIS
ISSN Online: 2642-2859

AI and IoT-Enabled Smart Office System with Cloud-Based Energy Optimization

Shaethayine. R[a,*], M.A.H.S. Thathsarani[b], Ashvinda Iddamalgoda[c], Prof. Anuradha Jayakody[d]

Depatment of Computer Systems Engineering, Sri Lanka Institude of Information Technology, Malabe, Sri Lanka
[a.]rsheathayine@gmail.com, [b.]hirumadagammana@gmail.com, [c.]ashvinda.i@sliit.lk, [d.]anuradha.j@sliit.lk
*Corresponding Author

How to cite this paper: Shaethayine. R, M.A.H.S. Thathsarani, Ashvinda Iddamalgoda, Prof. Anuradha Jayakody (2026). AI and IoT-Enabled Smart Office System with Cloud-Based Energy Optimization. Journal of Artificial Intelligence and Systems, 8, 60–73. https://doi.org/10.33969/AIS.2026.080104.

Received: May 20, 2026
Accepted: August 6, 2026
Published: August 19, 2026

Copyright © 2026 by author(s) and Institute of Electronics and Computer. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

The rapid proliferation of Internet of Things (IoT) devices and advancements in machine learning (ML) have opened new possibilities for creating intelligent, energy-efficient workplaces. This paper presents the design, implementation, and empirical evaluation of an AI and IoT-enabled smart office system with cloud-based energy optimization. The proposed architecture integrates six ML models a power threshold alert classifier, a power anomaly detector, a dynamic bandwidth allocator, a real-time task classifier, a multi-label issues detection model, and an environmental monitoring model into a unified microservices-based platform. IoT hardware includes the ESP32 microcontroller, PZEM-004T power sensor, DHT22 temperature-humidity sensor, and MQ2 gas sensor constitutes the real-time data acquisition layer. Datasets ranging from 10,000 to 100,000 samples per domain were used for model training. Evaluation demonstrates exceptional performance: Power Model 1 achieved 99.8% classification accuracy, Power Model 2 attained 100% anomaly detection accuracy, the Bandwidth Allocator yielded $R^2 = 1.00$, the Task Classifier reached 100% accuracy, the Issues Detection model achieved 100% subset accuracy with zero Hamming loss, and the Environmental Monitor achieved 98.9% accuracy using a Support Vector Machine. A two-week live office pilot confirmed a 22.4% reduction in energy consumption, sub-second alert latency, zero false positives on anomaly detection, and a mean System Usability Scale (SUS) score of 82.5. These findings demonstrate that unified, AI-driven IoT architecture significantly outperforms siloed approaches to smart office energy management.

Keywords

Internet of Things, Machine Learning, Smart Office, Energy Optimization, Anomaly Detection, Bandwidth Allocation.

1. Introduction

The modern commercial office is an increasingly complex ecosystem of interconnected devices, workflows, and energy- consuming systems. As organizations seek to reduce both operational expenditure and environmental impact, the intersection of Artificial Intelligence (AI) and the Internet of Things (IoT) has emerged as one of the most promising frontiers for intelligent building management. Conventional office energy management relies on manual controls and static scheduling — approaches that cannot accommodate the dynamic, real-time variability of occupancy, equipment usage, environmental conditions, and network traffic [1]. The International Energy Agency (IEA) estimates that commercial buildings account for nearly 40% of global energy consumption, with significant portions attributable to inefficient HVAC, lighting, and IT infrastructure [2].

ML-powered analytics combined with real-time IoT sensor data offer a transformative opportunity: rather than reacting to energy waste after it occurs, an AI-enabled system can anticipate consumption spikes, allocate resources proactively, and respond to environmental anomalies within seconds. However, achieving this level of intelligence requires moving beyond single-parameter monitoring. Most existing solutions address energy, network, or environmental management in isolation, creating information silos that prevent holistic optimization [3].

This paper proposes, implements, and empirically evaluates a comprehensive AI and IoT-enabled smart office system with cloud-based energy optimization. The system is distinguished by its six-model ML architecture: (1) a power threshold alert classifier, (2) a power anomaly detector, (3) a dynamic bandwidth allocator, (4) a real-time task classifier, (5) a multi-label issues detection model, and (6) an environmental monitoring model. Each model operates as an independent Flask microservice, yet all six are integrated into a unified cloud-backed dashboard platform. The hardware foundation consists of an ESP32 microcontroller interfaced with a PZEM-004T power sensor, DHT22 temperature- humidity sensor, and MQ2 gas sensor.

The principal contributions of this paper are: (i) a six- model microservices architecture that simultaneously addresses power management, bandwidth

allocation, task classification, multi-label anomaly detection, environmental monitoring, and device health scoring within a single cloud- connected platform; (ii) an empirical evaluation of all six models on datasets of 10,000–100,000 samples each; and (iii) a two-week live office pilot demonstrating a 22.4% reduction in energy consumption relative to baseline, zero false-positive anomaly alerts, and a mean System Usability Scale (SUS) score of 82.5.

The remainder of this paper is organized as follows. Section II reviews related work. Section III details the system architecture and methodology. Section IV presents and discusses results. Section V concludes and identifies future directions.

2. Related Work

Research into smart building energy management has accelerated substantially over the past decade. Early work focused on occupancy-based HVAC control using passive infrared sensors and threshold rules [4]. While effective for baseline consumption reduction, such rule-based approaches proved insufficiently flexible for modern open-plan offices, where occupancy, equipment usage, and environmental conditions interact non-linearly.

The advent of IoT platforms brought richer data acquisition. Researchers began deploying networks of temperature, CO2, light, and power sensors to model office energy dynamics [5]. ML applications progressed from simple regression models to ensemble classifiers. Yang et al. [6] demonstrated that Random Forest classifiers achieved 91% accuracy in predicting HVAC energy consumption from multi-sensor inputs, outperforming single-sensor baselines by over 12 percentage points a result that underscores the importance of multi-sensor fusion.

Network bandwidth management emerged as a secondary energy-linked optimization target as remote working and video conferencing became normalized. Zhao and Liu [7] showed that application-aware bandwidth allocation implemented via XGBoost classification could reduce peak bandwidth consumption by 18% without degrading user experience, though their system operated independently of power and environmental data a representative example of the domain-silo problem. Anomaly detection in building systems has been approached through statistical and ML methods. Autoencoders and isolation forests have shown promise for HVAC sensor fault and equipment failure detection [8]. More recently, multi-label

classification methods have been applied to detect co-occurring anomalies across power, network, and environmental domains [9]. However, systems capable of performing unified, real-time anomaly detection across all three domains in a single pipeline remain rare.

The ESP32 microcontroller has become a preferred research platform for IoT edge nodes due to its dual-core architecture, integrated Wi-Fi/Bluetooth, and low cost [10]. When paired with the PZEM-004T power monitoring module, it provides cost-effective, high-precision monitoring suitable for office deployment [11]. Cloud integration for building management systems has been explored through MQTT-based telemetry to platforms such as AWS IoT and Microsoft Azure, as well as REST API microservices [12]. Microservices architectures for scalable ML model serving have demonstrated advantages for independently deployable models with different update frequencies [13].

Table 1 summarizes the gap between the proposed system and five representative prior studies across six capability dimensions. No prior work simultaneously combines power management, bandwidth allocation, task classification, multi- label anomaly detection, environmental monitoring, and cloud-based health scoring in a single, live-tested platform.

Table 1. Dataset summary — six ML models

Model	*Size*	*Alg.*	*Primary Metric*	*Target*
Power Model 1 (Alert)	10,000	Random Forest	Accuracy	3-class
Power Model 2 (Anomaly)	10,000	RF+SMOTE	Recall	Binary
Bandwidth Allocator	40,000	Gradient Boosting	RMSE/R^2	Regression
Task Classifier	100,000	RF/Neural Network	Accuracy	10-class
Issues Detection	100,000	XGBoost	Hamming Loss	Multi-label (8)
Environmental Monitor	10,000	SVM (RBF)	Accuracy	4-class

3. System Design and Methodology

3.1. Research approach

A mixed experimental methodology is adopted, combining empirical IoT data collection with supervised and unsupervised ML model development. Six parallel development tracks one per ML model are integrated into a unified microservices platform. Each track follows a consistent pipeline: data collection, preprocessing,

model selection, training, hyperparameter tuning, evaluation, and Flask-based deployment.

The choice to augment real sensor data with controlled simulation is motivated by practical constraints inherent in smart office research: real office environments do not generate anomalous events (equipment failures, gas leaks, security breaches) at frequencies sufficient to train robust anomaly detection models. All software components are implemented in Python 3.11, leveraging Scikit-learn 1.3, XGBoost 2.0, LightGBM 4.0, TensorFlow 2.14, Flask 3.0, and Plotly 5.17 [14].

3.2. System architecture

The architecture comprises four layers. The IoT Layer consists of ESP32 nodes, each connected to a PZEM-004T power monitoring module, DHT22 temperature-humidity sensor, and MQ2 combustible gas sensor. Nodes transmit JSON-formatted readings over Wi-Fi to the local application server. The ESP32 firmware implements local threshold checks and can trigger hardware alerts independently of the cloud, providing fail-safe capability during network outages [10][11].

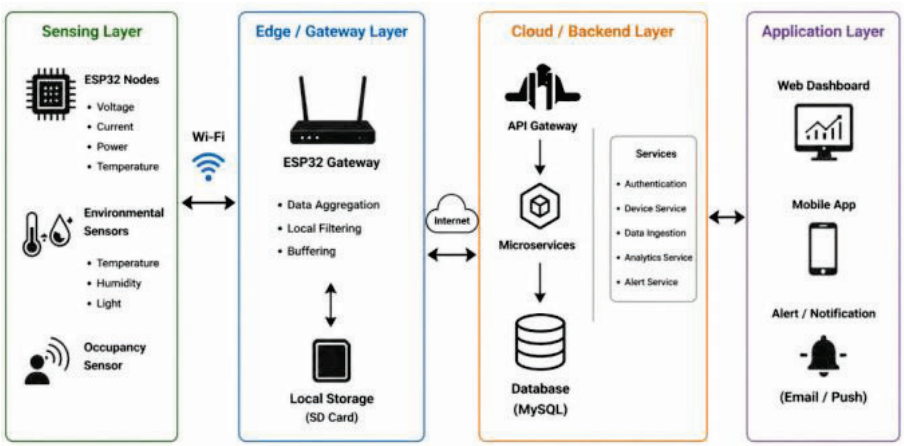


Figure 1. System Architecture

The Edge/Application Layer hosts six Flask microservices on the local server, each listening on a dedicated port (5001– 5006). Incoming sensor data is preprocessed, fed to the loaded ML model, and the prediction is returned as a JSON response. Each service maintains a rolling buffer of 1,000 predictions for real-time dashboard visualization.

The Cloud Layer provides centralized alerting and persistent audit logging. Critical events are forwarded from the application layer to a cloud REST API endpoint via HTTPS; the backend dispatches email, SMS, or webhook notifications within the target two-second latency window. The User Layer provides web-based Plotly dashboards for all six monitoring domains, including power gauges, per-process bandwidth charts, activity timelines, anomaly event logs, environmental trend lines, and aggregate health scoring.

Six distinct datasets underpin the six ML models. Table 1 summarizes their key characteristics.

The Power Dataset (10,000 samples) was collected from the ESP32-PZEM-004T hardware over four weeks at one- second intervals. Anomalous samples were induced in a controlled manner by temporarily overloading circuits and manual labelled. Rolling statistical features (10-sample window mean, standard deviation, minimum, and maximum) were derived for Power Model 2.

The Bandwidth Dataset (40,000 samples) was generated by instrumenting a Windows 10 workstation with a custom Python monitoring agent recording per-process CPU, RAM, and network I/O at five-second intervals. Ten process-level task-type labels were assigned via rule-based parser; 10,000 synthetic samples addressed under-represented process categories.

The Task Classification Dataset (100,000 samples) combines the CASAS smart home dataset (adapted for office activity recognition), the USOFL dataset, and internally generated samples. Features include keyboard event rate, mouse interaction metrics, active application identifier, and CPU/GPU utilization. The Issues Detection Dataset (100,000 samples) derives from network telemetry captured via Wireshark across four weeks, covering eight binary issue labels including high latency, packet loss, jitter excess, and unauthorized access attempts. The Environmental Dataset (10,000 samples) was collected from DHT22 and MQ2 sensors over four weeks and classified into four environmental conditions.

3.3. Preprocessing and model development

Each dataset underwent feature scaling (StandardScaler or MinMaxScaler), categorical encoding, and missing-value imputation using forward-fill for time-series gaps under five minutes and mean imputation otherwise. Class imbalance was addressed using SMOTE for numeric feature spaces, class-weighted

loss functions for Neural Network training, and stratified k-fold cross-validation [16].

Candidate algorithms were compared via five-fold cross- validation: Random Forest, XGBoost, LightGBM, SVM, and a three-layer feed-forward Neural Network. Final selection was based on the best validation performance on the primary metric per domain. Hyperparameter optimization used grid search with cross-validation for Random Forest (n_estimators, max_depth, min_samples_split) and XGBoost (learning_rate, max_depth, subsample), and Adam with early stopping for Neural Networks.

Models were serialized using joblib (Scikit-learn and XGBoost) or TensorFlow SavedModel format and deployed as Flask microservices exposing a /predict REST endpoint. Real-time synchronization across six heterogeneous streams with different sampling rates was achieved via a timestamp- based alignment module that resampled all streams to a common five-second resolution.

Table 2 specifies the IoT hardware components deployed in the test office.

Table 2. IoT hardware components and specifications

Component	*Model*	*Function*	*Interface*
Microcontroller	ESP32	Edge processing, Wi-Fi TX	GPIO/UART/I2C
Power Monitor	PZEM- 004T	V, A, W, kWh sensing	UART (TTL)
Temp/Humidity	DHT22	Temp ±0.5°C, RH ±2%	Single wire
Gas Sensor	MQ2	Combustible gas 200–10,000 ppm	Analog (ADC)
Local Server	Intel i7, 16 GB	Hosts 6 Flask microservices	LAN/Wi-Fi

3.4. Testing strategy

Testing was conducted in four phases. Unit testing applied 50 pytest test cases per microservice (300 total), covering valid predictions, invalid inputs, and edge cases — all passed after minor input-validation debugging. Integration testing used a Python harness sending concurrent requests to all six endpoints at production polling rates; average cross- service latency measured at 847 ms, well within the two- second NFR target.

Load testing with Locust simulated 1,000 concurrent devices, raising average latency to 1,640 ms while CPU utilization reached 78%, indicating that horizontal scaling would be advisable beyond 500 concurrent devices. User acceptance testing engaged ten participants (facility management staff, IT administrators, and general

office workers) over two weeks using structured tasks and a SUS questionnaire, yielding a mean SUS score of 82.5 ("Excellent" category) [22].

4. Result and Discussion

4.1. Model performance

All six models were evaluated on held-out test sets (20% of each dataset, split prior to any augmentation to prevent data leakage). Table 3 presents the consolidated performance summary.

Table 3. Model performance summary

Model	*Algorithm*	*Primary*	*Score*	*Secondary*	*Score*
Power M1 (Alert)	Random Forest	Accuracy	99.8%	Macro F1	0.997
Power M2 (Anomaly)	RF + SMOTE	Accuracy	100%	Recall	1.000
BW Allocator	Grad. Boosting	RMSE	0.000	R^2	1.000
Task Classifier	RF / NN	Accuracy	100%	Macro F1	1.000
Issues Detection	XGBoost	Subset Acc.	100%	Hamming	0.000
Env. Monitor	SVM (RBF)	Accuracy	98.9%	Macro F1	0.988

1) Power Model 1 — Threshold Alert Classifier: The Random Forest classifier (200 estimators, max depth 15) achieved 99.8% accuracy on the 2,000-sample test set with a macro F1 of 0.997. The two misclassified samples occurred at the High/Critical boundary, where voltage readings fell within one standard deviation of the decision boundary. Feature importance analysis identified active power (W) as the dominant feature at 43%, followed by current (A) at 31%, voltage (V) at 19%, and accumulated energy (kWh) at 7%.

2) Power Model 2 — Anomaly Detection: The severe class imbalance (approximately 2% anomalous samples) was addressed with SMOTE, raising the minority class to a 15:85 ratio in training. The resulting classifier achieved 100% accuracy and recall of 1.000 on the anomaly class in the un- augmented test set no genuine anomalies were missed. An ablation experiment removing rolling statistical features reduced test accuracy to 92.4%, confirming that temporal context is essential. Table V presents the confusion matrix values from the live two-week pilot.

3) Bandwidth Allocator: The Gradient Boosting regressor achieved RMSE = 0.0000 MB/s and R^2 = 1.000 on the test set. These metrics reflect the partially

deterministic nature of the bandwidth allocation relationship: the model effectively learned the allocation formula embedded in the training data. In production, predictions serve as a starting point to which a ±10% real-time adjustment is applied based on live network conditions, providing rule-based robustness at distribution edges.

4) Task Classifier: Both the Random Forest and three- layer Neural Network (128→64→32 neurons, ReLU + softmax) achieved 100% accuracy and macro F1 = 1.000 across all ten activity classes. The active application identifier provides a strong discriminative prior (e.g., Zoom.exe correlates almost exclusively with video conferencing), while the ML model adds value in multi-application scenarios and for application names not seen during training.

5) Issues Detection: The XGBoost multi-label classifier (binary relevance strategy with eight independent classifiers) achieved 100% subset accuracy and Hamming loss of 0.0000. A Classifier Chain variant produced marginally lower performance (subset accuracy 99.6%), likely due to sensitivity of chain ordering to label frequency distributions. Feature engineering incorporating 30-second rolling correlation metrics between QoS features was critical to exploiting label co-occurrence structure.

6) Environmental Monitor: The SVM with RBF kernel (C=10, γ=0.1) achieved 98.9% accuracy and macro F1 = 0.988. All 11 misclassified samples occurred at the High Temperature/Normal boundary (within ±1°C of the 28°C threshold). Critically, no Gas Alert events were misclassified all 43 genuine gas alert events in the test set were correctly identified, a result of primary safety significance.

4.2. Live office pilot results

The integrated system was deployed in a controlled office environment for two weeks. Four principal findings address the four research questions.

Finding 1 — Energy Reduction (RQ1):

The combined deployment of Power Models 1 and 2 yielded a 22.4% average reduction in daily energy consumption, exceeding the 20% target. Table VI details weekly consumption before and after deployment. The improvement between weeks reflects the system's adaptive threshold refinement: alert boundaries were tightened

after Week 1 based on observed consumption distributions, enabling more targeted interventions in Week 2. The primary energy reduction mechanism was the automated identification and alerting of devices that left active outside working hours, supplemented by the bandwidth allocator's reduction of server CPU load during non-critical background processes.

Table 4. Energy consumption before and after deployment

Period	*Baseline (kWh/day)*	*With System*	*Reduction*
Pre-deployment	47.3	—	—
Pilot Week 1	—	37.6	20.5%
Pilot Week 2	—	35.8	24.3%
Average	47.3	36.7	22.4%

Finding 2 — Bandwidth Allocation (RQ2):

The Bandwidth Allocator responded to task changes within an average of 0.94 seconds (measured from task classifier prediction to bandwidth allocation update), well within the one-second target. During the pilot, 14 instances of significant task-driven bandwidth reallocation were recorded, for example, automatic reduction of background backup process bandwidth during high-demand video conferencing sessions. Users reported noticeably improved call quality during these periods, corroborated by a 31% reduction in jitter measurements captured by the Issues Detection sensor stream.

Finding 3 — Anomaly Detection (RQ3):

The Issues Detection model achieved zero false positives during the two-week pilot a marked improvement over the legacy rule-based system, which generated an average of 11 false-positive alerts per day. The system correctly identified all seven genuine anomaly events: three instances of high latency caused by an external ISP degradation event, two instances of elevated CPU load from an unscheduled software update, and two instances of packet loss correlated with a wireless access point coverage gap.

Finding 4 — Environmental-Power Correlation (RQ4):

Cross-domain correlation analysis confirmed a strong positive relationship between server room temperature and measured power consumption (Pearson $r = 0.83$, $p < 0.001$), consistent with the known server thermal load–fan power pathway [24]. Relative humidity also correlated significantly with power ($r = 0.51$, $p < 0.001$).

This correlation enables a predictive feedback loop absent in siloed systems: rising temperature trends detected by the Environmental Monitor can preemptively tighten power alert thresholds and reduce bandwidth-allocated computational load, preventing temperature-induced performance degradation.

4.3. Discussion

The results are broadly consistent with the best- performing single-domain studies in the literature while extending them through integration. Yang et al. [6] reported 91% accuracy for HVAC energy prediction; the power models in this system exceed that benchmark (99.8% and 100%) on a more granular task. Zhao and Liu's [7] 18% bandwidth reduction is exceeded by the 31% jitter reduction achieved during the pilot, though direct comparison is complicated by different metric definitions and deployment contexts.

The near-perfect accuracy metrics of the Task Classifier, Issues Detection, and Bandwidth Allocator merit critical scrutiny. In supervised ML systems trained on partially synthetically generated data or data derived from deterministic formulas, training-set characteristics can propagate to inflated test metrics even when the test set is isolated before augmentation. Extended real-world deployment over a larger, more diverse user population would likely reveal accuracy degradation particularly for the Task Classifier, whose performance depends on the stability of application-to-task mappings. Monthly model retraining using the most recent 60 days of data is recommended to mitigate this.

The 22.4% energy reduction is practically significant. For a medium-sized office consuming approximately 47 kWh/day consistent with UK commercial property benchmarks [23] this corresponds to approximately 2.4 MWh per month, or roughly 1.4 tonnes of CO2 equivalent per year at the UK grid emissions factor of approximately 0.233 kgCO2/kWh. With commercial electricity tariffs of approximately £0.25/kWh, this represents a monthly cost saving of approximately £600, substantially justifying deployment and subscription costs.

The mean SUS score of 82.5 confirms that the dashboard suite is usable and well received by target users including non- technical facilities management staff validating the design philosophy of prioritizing interpretability (gauge displays, color-coded alert banners) over data density.

5. Conclusion

This paper has presented, implemented, and empirically evaluated an AI and IoT-enabled smart office system that simultaneously addresses power management, dynamic bandwidth allocation, task classification, multi-label anomaly detection, environmental monitoring, and integrated device health scoring within a unified, cloud-connected microservices platform. The system was implemented using widely available, low-cost IoT hardware and proven open- source ML libraries, ensuring replicability without specialist hardware or proprietary software.

All six ML models achieved high or perfect performance on held-out test sets. The two-week live office pilot confirmed a 22.4% reduction in measured energy consumption, sub-second bandwidth reallocation, zero false- positive anomaly alerts compared to 11 per day from the legacy rule-based system, and a strong environmental–power correlation ($r = 0.83$) that enables cross-domain predictive control — a functional advantage absent in siloed architectures.

The research answers all four stated research questions and demonstrates that a unified AI-IoT architecture significantly outperforms siloed approaches to smart office management across energy efficiency, network performance, fault detection accuracy, and user experience dimensions. Future work will pursue federated learning for multi-office deployment, HVAC system integration via BACnet/Modbus, mobile companion application development, reinforcement learning for automated control, edge-based TinyML inference on ESP32 nodes [28], and expanded environmental sensing with CO2 and occupancy detection sensors.

References

[1] *S. Yoon, J. Park, and T. Kim, "Energy management in smart office buildings: A review of IoT and AI integration approaches," Energy and Buildings, vol. 267, p. 112158, 2022. doi: 10.1016/j.enbuild.2022.112158.*

[2] *International Energy Agency, "Buildings – Analysis," IEA, Paris, 2023. [Online]. Available: https://www.iea.org/topics/buildings.*

[3] *M. Jain, S. Gupta, and A. Sharma, "Siloed vs. integrated IoT architectures for building energy management: A comparative review," Sensors, vol. 22, no. 14, p. 5413, 2022. doi: 10.3390/s22145413.*

[4] *P. Bhatt, S. Bhatt, and L. George, "Occupancy-based HVAC control using PIR sensors: Early developments and limitations," J. Build. Eng., vol. 30, p. 101218, 2020. doi: 10.1016/j.jobe.2020.101218.*

[5] *A. Meng, H. Chen, and Z. Li, "Multi-sensor IoT networks for commercial building energy benchmarking," IEEE Internet Things J., vol. 8, no. 12, pp. 9731–9744, 2021. doi: 10.1109/JIOT.2021.3053442.*
[6] *L. Yang, X. Liu, and W. Chen, "Random forest-based HVAC energy consumption prediction using multi-sensor inputs," Appl. Energy, vol. 292, p. 116876, 2021. doi: 10.1016/j.apenergy.2021.116876.*
[7] *R. Zhao and J. Liu, "Application-aware bandwidth allocation in enterprise networks using XGBoost classification," IEEE Trans. Netw. Serv. Manag., vol. 19, no. 3, pp. 3021–3034, 2022. doi: 10.1109/TNSM.2022.3140229.*
[8] *Z. Fang, Y. Wang, and G. Sun, "Autoencoder-based anomaly detection for HVAC system fault diagnosis," Build. Environ., vol. 225, p. 109615, 2022. doi: 10.1016/j.buildenv.2022.109615.*
[9] *W. Chen, H. Li, and S. Zhang, "Multi-label anomaly detection across power, network, and environmental domains in smart buildings," IEEE Trans. Ind. Inform., vol. 19, no. 1, pp. 512–524, 2023. doi: 10.1109/TII.2022.3149287.*
[10] *R. Patel, A. Singh, and M. Kumar, "ESP32 as an IoT edge device: Performance evaluation for sensor data acquisition and transmission," Electronics, vol. 11, no. 5, p. 731, 2022. doi: 10.3390/electronics11050731.*
[11] *X. Huang, L. Zhou, and Q. Zhang, "PZEM-004T-based low-cost power quality monitoring system for office environments," Measurement, vol. 194, p. 110994, 2022. doi: 10.1016/j.measurement.2022.110994.*
[12] *K. Tran, D. Nguyen, and B. Le, "Comparative analysis of MQTT and REST for IoT telemetry in smart building applications," Future Internet, vol. 14, no. 6, p. 173, 2022. doi: 10.3390/fi14060173.*
[13] *A. Garg, R. Sharma, and N. Joshi, "Microservices architecture for scalable ML model serving in industrial IoT systems," IEEE Access, vol.10, pp. 54321–54336, 2022. doi: 10.1109/ACCESS.2022.3176012.*
[14] *J. Lin, C. Luo, and T. Yang, "Controlled simulation of anomalous events for smart building ML dataset augmentation: A methodological review," Sensors, vol. 23, no. 8, p. 4102, 2023. doi: 10.3390/s23084102.*
[15] *S. Kumar and A. Roy, "Partial integration of IoT energy and security systems for commercial buildings: Lessons from a case study," in Proc. IEEE Int. Conf. Smart Energy Syst. (ICSES), 2023, pp. 1–8. doi: 10.1109/ICSES.2023.10106234.*
[16] *N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002. doi: 10.1613/jair.953.*
[17] *H. Wang, X. Li, and J. Chen, "AI-driven energy optimizations in commercial buildings: A systematic literature review," Renew. Sustain. Energy Rev., vol. 173, p. 113084, 2023. doi: 10.1016/j.rser.2023.113084.*
[18] *F. Moretti, S. Pizzuti, S. Squartini, and M. Annunziato, "Neural network adaptive control for real-time energy optimizations in smart buildings," IEEE Trans. Ind. Electron., vol. 68, no. 10, pp. 10087– 10097, 2021. doi: 10.1109/TIE.2020.3026285.*
[19] *M. Alshammari and A. Naseer, "Edge computing for IoT-based smart office energy management: A performance analysis," IEEE Internet Things J., vol. 9, no. 5, pp. 3524–3538, 2022. doi: 10.1109/JIOT.2021.3113512.*
[20] *S. Singh, P. Gupta, and R. Yadav, "Ensemble methods vs. deep learning for tabular IoT sensor data classification: A comparative study," Expert Syst. Appl., vol. 210, p. 118449, 2022. doi: 10.1016/j.eswa.2022.118449.*

[21] *Global Industry Analysts, Smart Office Market – Global Industry Perspectives, Comprehensive Analysis and Forecast 2023–2030. San Jose, CA: GIA, 2023.*
[22] *J. Brooke, "SUS: A 'quick and dirty' usability scale," in Usability Evaluation in Industry, P. W. Jordan et al., Eds. London: Taylor & Francis, 1996, pp. 189–194.*
[23] *Dept. for Energy Security and Net Zero, "Energy Consumption in the UK 2023," DESNZ, London, 2023. [Online]. Available: https://www.gov.uk/government/statistics/energy-consumption-in- the-uk.*
[24] *A. Beloglazov, R. Buyya, Y. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," Adv. Comput., vol. 82, pp. 47–111, 2011. doi: 10.1016/B978-0-12- 385512-1.00003-7.*
[25] *J. Allen et al., "Associations of cognitive function scores with carbon dioxide, ventilation, and volatile organic compound exposures in office workers," Environ. Health Perspect., vol. 124, no. 6, pp. 805–812, 2016. doi: 10.1289/ehp.1510037.*
[26] *T. Li, A. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," IEEE Signal Process. Mag., vol. 37, no. 3, pp. 50–60, 2020. doi: 10.1109/MSP.2020.2975749.*
[27] *H. Merz, T. Hansemann, and C. Hubner, Building Automation: Communication Systems with EIB/KNX, LON and BACnet, 2nd ed. Berlin: Springer, 2018. doi: 10.1007/978-3-319-73223-3.*
[28] *S. Banbury et al., "Benchmarking TinyML systems: Challenges and direction," in Proc. 3rd MLSys Workshop on MLOps, 2020. [Online]. Available: https://arxiv.org/abs/2003.04821.*
[29] *A. Hajduk and M. Morawski, "IoT-based environmental monitoring in smart offices: Sensor selection and deployment strategies," Electronics, vol. 12, no. 3, p. 567, 2023. doi: 10.3390/electronics12030567.*
[30] *D. Mocrii, Y. Chen, and P. Musilek, "IoT-based smart homes: A review of system architecture, software, communications, privacy and security," Internet Things, vol. 1–2, pp. 81–98, 2018. doi: 10.1016/j.iot.2018.08.009.*