Deep Learning-Driven Anomaly Detection for Green IoT Edge Networks

Ahmad Shahnejat Bushehri, Ashkan Amirnia, Adel Belkhiri, Samira Keivanpour, Felipe Gohring de Magalhães, Gabriela Nicolescu

***Abstract*—The widespread use of sensor devices in IoT networks imposes a significant burden on energy consumption at the network's edge. To address energy concerns, a prompt anomaly detection strategy is required on demand for troubleshooting resource-constrained IoT devices. It enables devices to adapt their configuration according to the dynamic signal quality and transmission settings. However, obtaining accurate energy data from IoT nodes without external devices is unfeasible.**

This paper proposes a framework for energy anomaly detection of IoT nodes using data transmission analysis. We use a public dataset that contains peer-to-peer IoT communication energy and link quality data. Our framework first utilizes linear regression to analyze and identify the dominant features of data communication for IoT transceivers. Later, a deep neural network modifies the gradient flow to focus on the dominant features. This modification improves the detection accuracy of anomalies by minimizing the associated reconstruction error. Finally, the energy stabilization feedback provides nodes with insight to change their transmission configuration for future communication.

The experimental results show that the proposed approach outperforms other unsupervised models in anomalous energy detection. It also proves that redesigning the conventional loss function by enhancing the impact of our dominant features can dramatically improve the reliability of the anomaly detection method.

***Index Terms*—Transmission protocol, Anomaly detection, Edge networks, Unsupervised learning, IoT, Energy consumption, Power consumption, Link quality, Deep learning, Low power, Wireless sensor networks, Smart city**

I. INTRODUCTION

AN Internet of Things (IoT) scenario on edge is defined through a set of wireless sensor networks connecting a collection of widely distributed monitoring sensor devices (SD) [1]. This scheme would collect and send the desired data to the edge servers for further processing and decision-making [2]. Such tiny sensor devices feature a severe constraint on the hardware resource, such as stripped RAM size, low-rank CPU/GPU capability, and limited energy harvesting [3]. However, they are well-adopted in autonomous systems for steady data collection, stable transmission [4], continuous diagnosis [5], and real-time analysis [6]. Energy consumption is counted as one of the most challenging constraints for SD service provisioning. Energy over-consumption occurs in cases like amplifying the transmission power for fading shadowing avoidance. Similarly, when SDs are located in remote areas and distant from gateways, their vulnerability to increased path loss becomes more pronounced, particularly in terms of energy efficiency. Hence, energy optimization techniques are required to increase the battery lifetime of SDs and maintain high-efficiency levels [7].

Most software-layer energy optimization models perform at the levels of systems, architecture, or circuits [8] [9]. The outcome of these energy minimization approaches is categorized into multiple groups for embedded computing devices [10]. The first is called Dynamic Power Management (DPM) [11]. DPM temporarily halts any idle element of the system to avoid wasting energy, like processing cores. In the second group, Dynamic Voltage-Frequency Scaling (DVFS) permits CPUs to switch between different frequency-voltage levels to scale down energy consumption [12]. To save a significant amount of energy expended on generic tasks on the third group, CPUs are customized to meet the requirements of the tasks on a multiprocessor system-on-chip [13]. The same customization happens for the fourth group but for the cache-based memory access [14]. By mapping tasks to different processors, the fifth group performs load balancing across all the cores which results in the utilization enhancement of processing elements on a multiprocessor system-on-chip [15].

The mismanagement of energy consumption can result in over-heat generation. The non-uniform heat generation leads to spatial temperature gradients over the whole chip. The more the load on a system caused by concurrent processing, the higher temporal heat generation/dissipation occurs. This may impose the temporal thermal gradients on a small area on the chip. The spatial temperature gradients and temporal thermal gradients besides the thermal cycles are three main factors that degrade the performance and the efficiency of a system during its lifetime. Ignoring overheating in edge nodes may also cause significant performance, durability, and reliability loss. For example, raising the temperature by ten to twenty degrees in metallic materials would shorten the useful life of the components by sixteen times [16].

However, applying the discussed solutions on a peer-to-peer network is not highly effective in reducing energy consumption for SDs and alleviating the consequences. These approaches do not consider transmission energy consumption as the primary cause of battery depletion for SDs in a wireless sensor network [17]. Although sensor devices are typically designed as a single processor board, most discussed approaches are devised for multiprocessor system-on-chips. For example, over-heat generation is much more prone to happen in multiprocessors than single processor SDs as the former suffers from non-homogeneous heat dissipation in its processing elements.

SDs can be classified into normal, sink, and gateway nodes. One of the main advantages of these resource-constrained devices located in hostile environments [18] is forwarding the collected data to the gateway nodes [19] for further analysis and decision-making (DM) [20]. The nodes with a higher rate of forwarding DM data are more susceptible to energy failure. By collecting energy and other transmission data (link quality parameters and configuration settings), an AI-based model can detect anomalies in energy behavior. As supervised learning solutions require data lebeling for energy over-consumption detection [21], they are not practically feasible for continuous learning in IoT domains. Consequently, there is a need to investigate if other energy-saving approaches (including unsupervised models and routing protocols) may prolong the survival time of the whole network reliably [22].

The principal contributions of our study can be summarized as follows.

- Propose a new energy-aware framework for the detection of energy consumption anomalies that might affect IoT nodes' lifetime in a wireless sensor network. This framework collects peer-to-peer communication energy and link quality data and uses a deep-learning approach to detect the abnormal energy behavior of the SDs;
- Propose a learning solution for anomaly detection based on an unsupervised algorithm. We design a new loss function for the conventional variational autoencoder by analyzing the feature importance of the data using a linear regression model;
- Evaluate the proposed approach by using a public dataset. Our results show that our approach achieves high-performance metrics and less intrusive execution time compared to other alternatives;
- Implement the proposed approach with other baseline clustering algorithms and unsupervised deep-learning models, including multi-layer VAEs, to make a comparison between their effectiveness and that of our proposed solution.

This paper is organized as follows. In the following section, we introduce energy-saving approaches at the network level. The problem statement and research question are presented in Section III. Data and theoretical analysis are provided in Section IV. The methodology and the details of the energy-aware framework are explained in Section V. The results of the experiments are described in Section VI. Finally, conclusions are drawn in Section VII.

II. Background and Related Works

Minimizing energy footprint can be performed at the node and cloud levels. At the node level, depending on the communication scenario, such as the location of nodes and the signal quality, data transmission is the primary cause of energy consumption of an IoT node in wireless sensor networks [23]. At the cloud level, data centers as powerful processing units are the decisive elements in the cloud computing domain. They host several power supply blocks, control systems, data communication, storage, and cooling systems, in addition to security components. Evaluating energy consumption in such complex systems is challenging as it requires a deep investigation of various connected elements [24]. We present the above-mentioned energy-saving approaches at the node and cloud levels in the following sections.

A. Node-based solutions

For IoT sensor nodes, extensive resource exhaustion occurs in transmit (TX) and receive (RX) states [25]. Several solutions are offered to remedy the high energy consumption rate and increase the life expectancy for sensor nodes in an IoT scenario. Without working on reliable energy-saving strategies for IoT nodes, a considerable strain will be put on the energy harvesting sources to provide sufficient supply for operational activities. There are four primary categories for network-based energy-saving approaches on IoT nodes: sleep/wake-up protocols, radio optimization, energy-efficient routing, and data reduction [26]. As we focus on dynamic link quality and transmission settings in this work, the first two approaches are investigated here to represent the role of protocols in the transmission energy consumption of a node.

1) Radio Optimization: Radio optimization is a set of approaches applied to a radio module as the most energy-hungry element in a sensor network to optimize the required energy for data communication. This type of radio optimization manages the energy consumption regime by manipulating the radio parameters. To this end, several optimization techniques for power transmission, modulation, and antenna direction are suggested. Moreover, a cooperative communication strategy and cognitive radio approach deliver substantial improvement to suppress the appetite for the remained energy in wireless sensor communication [27]. The cognitive radio, a prominent strategy against energy over-consumption in retransmissions (caused by packet loss), enables a transceiver to distinguish and avoid the occupied channels and access the vacant channels for data communication. Based on channel states, IoT nodes can intelligently pick a proper channel in the wireless spectrum and accordingly regulate the TX/RX parameters [28]. By comparing the cognitive-radio-enabled sensor networks with the classic ones, researchers in [29] indicated a 13% improvement in the network's lifespan. In cooperative communication strategies, a virtual multi-antenna array is achieved by using multiple single-antenna nodes and sharing antennas with them, which results in a remarkable enhancement of the received signal strength (RSS) [30]. Evaluating these strategies shows that a large amount of energy could be saved for contributors in a wireless sensor network. This regime designates a set of nodes as relays and uses them to distribute the communication load by reducing the distance between IoT nodes and the gateway. Hence, the required energy for data collection through the gateway would be cut significantly, and the lifetime of the sensor network would be extended.

Selecting proper antennas affects the energy performance of sensor networks significantly. For example, replacing an omnidirectional with a directional antenna limits the unwanted interference caused by retransmissions [31]. The considerably enhanced energy comes from the fact that if a transmission is done toward a specific direction, the neighbor communications could also be accomplished without interfering with others [32]. Although the advantages of removing omnidirectional antennas have been demonstrated, there are many practical difficulties in direction adjustment and identifying proper parameters for directional antennas. One of the solutions for the radio modulation parameter selection could be the modulation optimization strategy that contributes to energy minimization [33] [34]. By carefully setting the modulation parameters in a wireless sensor network, a balanced compromise between the transmission power and the inter-node distance can be devised to regulate an appropriate circuit/transmission power consumption. In radio communications, the transmission power is managed using the Dynamic/Transmit Power Control (DPC/TPC) mechanism [35]. While maintaining the necessary link quality parameters, such as the Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI), DPC reduces the power of a radio transmitter. By avoiding interference, it results in network life span extension as well [36]. However, applying radio optimization techniques such as adaptive power adjustment at a high rate makes communications vulnerable to packet loss.

2) Sleep/Wake-Up Protocols: A practical solution for prolonging the life span of event-driven wireless sensor networks is to change the behavioral pattern of IoT nodes for data communication with the edge. The idea behind behavioral tailoring in event-driven networks is to intermittently let the IoT nodes switch to sleep/awake mode. While an event is scheduled, the nodes would be in the awake state (radio on) just before data communication is initiated, and they would be switched back to sleep mode (radio off) as soon as the data stops flowing (no-event availability). Offering such a mechanism effectively contributes to energy footprint optimization in a resource-constrained edge network. For establishing a connection in this scenario, the sender and receiver should acquire a priori knowledge about the sleep-wake schedule of the neighbors (synchronization step). Given that information, the endpoint switches to the active mode right before the pre-determined moment and is subsequently involved in the data communication. To conduct the synchronization step, a topology control protocol can be used while adjusting the communication duty cycle between individual IoT nodes. The duty cycling technique offers an energy efficiency scheme by enabling nodes to switch between sleep and awake modes at regular intervals [37]. As the amount of consumed energy in awake mode gets double compared to that of sleep mode [38], a tuned duty cycle makes the sleep period extended while keeping the idle listening mode remarkably limited [39]. Duty cycling turns the radio on by detecting the preamble of a low-level carrier. Optimizing the energy consumption could be carried on through the topology control techniques (connectivity- and location-oriented) [40] [41], as they minimize the number of essential IoT nodes to meet the network connectivity and coverage requirements. The connectivity-oriented protocols for wireless sensor networks toggle IoT nodes between ON and OFF states dynamically to verify inclusive coverage and guarantee connectivity among nodes. On the other hand, according to the physical location of the IoT nodes, location-oriented protocols decide which nodes are qualified to be put in ON mode and when this transition takes place in a wireless sensor network to ensure network connectivity and coverage. While this approach reduces energy consumption considerably, it suffers from a critical weakness. Real-time data collection cannot be performed using this technique as it requires a high-duty cycle of an IoT node and, subsequently, results in high energy consumption.

B. Cloud-based solutions

Most proposed approaches addressing high energy consumption are grouped into idle server shutdown and workload consolidation. However, machine-learning-based solutions have recently seen an upward trend in energy optimization. M. Demirci in [42] presented the produced heat, virtual machine placement, task scheduling, and power distribution as the main contributors to power consumption, while the environmental control overhead and processing overload control are considered side effects. However, resource prediction without a priori knowledge of allocated tasks to the data centers usually comes with a large error margin. Thus, an efficient predictive model can significantly enhance energy consumption by providing an optimized allocation approach. DejaVu [43] is presented as a cloud-based resource management system. The authors made a classification for load consolidation through a supervised machine-learning approach. As a result, the system started to be trained by analyzing the allocation process for varying workloads. After this step, DejaVu has already learned to classify any unseen workload in two ways. It checks if the load is similar to any of those previously seen or not. If so, the exact allocation applies. Otherwise, it is treated as a new sample to be learned. As another load forecasting strategy, a neural-network-oriented predictor is employed in [44] to forecast how large the cloud will be loaded in the future. According to that, the unnecessary running machines would be switched off to prolong the total uptime of the system. Such green scheduling solutions can keep the same productivity with fewer servers while consuming less than 60% of energy compared to the standard scenario. Other works [45] anticipate running tasks' future energy footprint

TABLE I: Relevant energy saving mechanisms in IoT domain

	Approaches	Methodology	Strategy	Articles
Energy Saving Mechanisms	Node_based	Radio Optimization	Cognitive Radio	[27], [28], [29]
			Cooperative Communication	[27], [30]
			Directional Antenna	[31], [32]
			Modulation Optimization	[33], [34]
			Transmission Power Control	[35], [36]
		Sleep/Wake-Up Protocols	Duty Cycling	[37], [39]
			Topology Control	[40], [41]
	Cloud_based	Node Sleeping	Unsupervised Learning	[46]
			Supervised Learning	[43], [58], [59], [60], [61]

and service level agreement metrics like response time using supervised machine learning. The insightful feedback provided by such estimation helps the maintenance team devise energy-efficient task scheduling for high-demanding data centers. Applying unsupervised machine learning (ML) [46] could estimate the required resources by virtual machines. By evaluating the resource estimation, this approach reduces the consumed energy in data centers by turning unnecessary machines into sleep mode. A centralized approach in [47] is used for resource allocation using Q-learning in Heterogeneous cloud radio access networks. Despite a considerable energy budget saving by this framework, their analysis reports that the interference mitigation is met and the quality of service is acceptably preserved among users. Though the data analytics approaches discussed so far are related only to data centers, leveraging machine learning decision-making to resource-constrained edge devices is also a high-demanding trend. Due to the limited hardware/software capabilities of edge devices for IoT applications, mapping large-scale ML algorithms is not practical. However, an energy-efficient in-memory computing kernel is proposed for linear classification [48]. They use a split-data-aware mechanism to adjust voltage, process, and temperature changes. This results in an admissible balance between energy efficiency and accuracy features. A summary of the discussed energy-saving mechanisms is presented in Table I.

Most introduced energy over-consumption detection approaches use external devices such as smart plugs [49] [50] [51] or are designed for smart buildings rather than computing devices [52] [53] [54]. Some other strategies follow general-purpose models that do not meet the constraints for edge nodes or use dynamic energy and thermal models with high overhead [55] [56] [57]. Edge-based energy modelings are mostly application-specific and may not necessarily target the overall energy consumption of the edge node. In some others, the energy requirements are known in advance based on contextual information such as location, time, and device capabilities. In others, the energy-controlling regime will mainly be deployed unacceptably late. Through our work, we target the limitations of the existing energy anomaly detection models to devise a non-intrusive and affordable mechanism to reduce the energy footprint of an edge network.

III. Problem Statement

A pattern in collected data belongs to an anomaly category if it does not conform to the defined expected behavior compared to other experiments [62]. In practice, an extensive range of anomalies may affect the energy efficiency of a wireless sensor network or even expose the IoT nodes to complete failure. Early anomaly detection strategies for IoT nodes are enabled by either internal or external data collection. Internal data collection consists of continuous extraction of granular tracing data [63] [64] [65] [66] out of IoT nodes along with gauged energy consumption so that abnormal energy patterns can be identified and avoided in future scenarios. On the other hand, external data collection consists of monitoring the data communication among edge elements.

Broadly speaking, detected anomalies can be categorized into two groups: point and contextual anomalies [67]. The former discusses a rare individual experiment that completely diverges from the energy of the majority of other observations. An example of this type of generated anomaly is a temporary sensor glitch that was raised from a bug in running software. On the other hand, contextual anomalies deal with a set of improper observations that occur in a particular time window, considering that their values do not reveal their nature of abnormal behavior at first glance. A possible verification process is to detect inappropriate patterns by comparing and correlating the patterns of data in time-related periods (day hours vs. night hours, etc.) for a long run [68].

It is worth mentioning that there are also cases in which a node works in an underconsumption state of its energy. Although we believe that this scenario could also be an anomaly, we decided to focus only on the overconsumption of energy to delimit the scope of our research in this paper. We believe that IoT nodes and in general sensor devices are designed to spend most of their lifetime in a standby or sleep mode for an unknown time period without intending to start any communication with other elements. Hence, they often do not spend energy on data transmission. Unlike obtaining a clear threshold for overconsumption, there is no valid minimum threshold for underconsumption in WSN as explained. Accordingly, if an IoT node consumes energy much less than others, there is no clear threshold-based strategy to detect an anomaly just through transmission data analysis. Capturing the anomalies in energy underconsumption in this case, may require more information regarding the internal execution trace of the system such as system call, resource utilization, and timing analysis to unveil the potential failure/malfunction of software or hardware components. To this end, our concern in this paper is to avoid overconsumption and prolong the lifetime of a node as there are cases in which inaccurate node configuration may reduce its operational lifetime by a hundred times [69].

IV. Data Description and Mathematical Analysis

In this work, we use a public dataset[1] which contains peer-to-peer IoT communication energy and link quality data. The data is collected and used by [70] [71] [72] to provide autonomous configuration of communication systems, sustainable modular solutions, and heterogeneous communication schemes for smart IoT nodes in various environments. The data collection occurs in a context where several IoT nodes are distributed around the experimental environment. IoT nodes with variant communication protocols for data transmission surround a unique gateway. Since the nodes are spread randomly around the gateway, the distance to the gateway varies in line of sight per node. Using variable transmission power and protocol for each node mapped in diverse geospatial locations results in various energy consumption at the transmitter's side and different RSS indicators at the gateway. The dataset is composed of 18448 observations, while each entry includes nine distinctive features as presented below:

- X position: X coordinate relative to the location of the gateway (0,0)
- Y position: Y coordinate relative to the location of the gateway (0,0)
- Scenario: Characteristics of the environment in which the transmission is carried out (i.e., indoors and outdoors)
- Distance: The distance in meters within which the gateway can be reached in line of sight
- Obstacles: The number of interfering obstacles placed between an IoT node and the gateway in line of sight
- Protocol: The transmission protocol used by each IoT node
- Power: The transmission power used by an IoT node
- Energy: The energy used by an IoT node to carry out the transmission
- RSSI: The received signal strength indicator at the gateway

Although in this framework all the connections are established with only one gateway, it could be easily extended to an actual scenario with multiple gateways. In that case, for each individual transmission initiated between a certain node and the counterpart gateway, our proposed framework can apply diagnostic troubleshooting actions separately.

[1] https://www.kaggle.com/datasets/andregloria/p2p-iot-communication-energy-and-link-quality.

A. Feature Correlation

To display a clear picture of co-linearity between different features of observations, a correlation matrix heatmap is depicted in Fig. 1. Each cell in this matrix contains the correlation coefficient. Blue reflects positive, and red shows a negative correlation. The darker the color, the larger the correlation coefficient magnitude. Targeting *Energy*, the two most correlated features are *Protocol* and *Power*. In addition to the practical correlation investigation done by the covariance matrix, a study on theoretical correlation dependency is also required to find how transmission-oriented features, including *Protocol*, *Power*, and *RSSI*, affect the *Energy* consumption of IoT nodes.

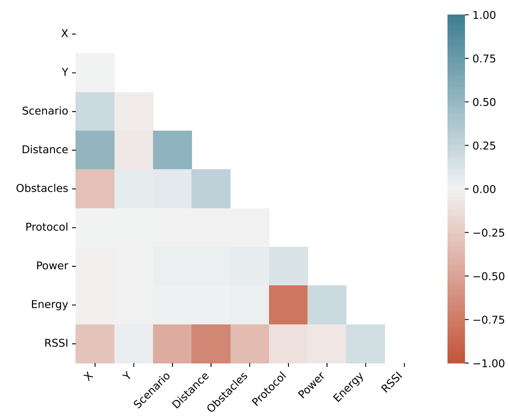


Fig. 1: Covariance matrix

B. Mathematical Analysis

In this part, a mathematical energy analysis for our model is provided to justify investigating the possible role of transmission parameters in energy consumption. Here, three terms are introduced during a machine cycle to represent the amount of time spent on each machine state. First, the amount of time for an IoT node in the awake state with the radio on is defined as $\text{Time}_{\text{active}}$. Besides that, the period just before initializing data communication is called $\text{Time}_{\text{sleep}}$ in which the radio turns off. The overall machine cycle ($\text{Time}_{\text{Total}}$) is hinged on the $\text{Time}_{\text{active}}$ in conjunction with $\text{Time}_{\text{sleep}}$ in a wireless sensor network [73].

$$\text{Time}_{\text{Total}} = \text{Time}_{\text{active}} + \text{Time}_{\text{sleep}} \tag{1}$$

Fig. 2 depicts a wireless sensor network on the edge side. For a machine cycle, the overall energy consumption for data communication in a wireless sensor network consists of $\text{Energy}_{\text{active}}$ and $\text{Energy}_{\text{sleep}}$. The first part of $\text{Energy}_{\text{Total}}$ is estimated based on the used energy for data transmission, reception, and task processing on the micro-controller unit (MCU) in the period of $\text{Time}_{\text{active}}$. The second part is estimated by the amount of energy consumed by all operating units like the sensors, MCU, and transceivers while the IoT

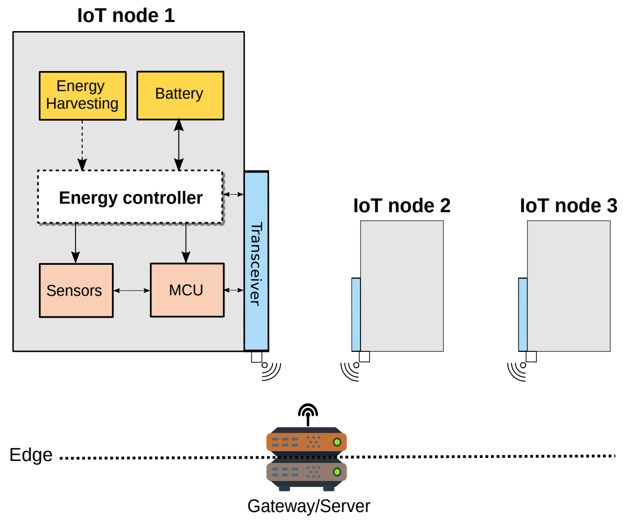


Fig. 2: IoT nodes communicating with the gateway at the edge

node is in sleep mode during $\text{Time}_{\text{sleep}}$. The parameters in Equations 1 and 2 are summarized in Table II.

$$\text{Energy}_{\text{Total}} = \overbrace{\underbrace{E_{\text{sensor\_ON}}}_{\substack{P_{\text{sensor\_ON}} \\ \times T_{\text{sensor\_ON}}}} + \underbrace{E_{\text{micro\_ON}}}_{\substack{[(P_{\text{micro}}+P_{\text{RX}})\times T_{\text{RX}}] \\ +[(P_{\text{micro}}+P_{\text{TX}})\times T_{\text{TX}}]}}}^{E_{\text{active}}} + \overbrace{P_{\text{sleep}} \times \text{Time}_{\text{sleep}}}^{E_{sleep}} \tag{2}$$

In general, the signal broadcasted by the source is affected by multiple environmental factors associated with the geospatial information of the transmission's location. For example, RSSI indicates the estimated level of a radio signal power as it is delivered to the receiver endpoint. Since the T-R pathway might comprise multiple obstacles in between, the path loss behavior roughly follows the log-normal shadowing rather than deterministic modelings such as the ground reflection (Two-Ray) or log-distance approaches [74]. Besides, the signal power behavior does not conform with the log-distance path loss model along the T-R range. Consequently, considering the random variations in path loss for IoT nodes in any T-R range, the log-normal shadowing in Equation 3 estimates the RSSI while avoiding the symmetric path loss values [75].

$$\text{RSSI} = P_{TX} - \underbrace{PL_d}_{PL_{d_0}+10\eta\log_{10}\left(\frac{d}{d_0}\right)+X_\sigma}, \tag{3}$$

Where, PL_d stands for a reference path loss in the range of random d meters from the transmitter, η is the path loss rate at which the RSSI drops with the distance d. X_σ in this equation defines the $X \sim \mathcal{N}(\mu, \sigma^2)$ with zero mean ($\mu = 0$) and standard deviation of σ.

The effect of transmission power on the communication range and the energy consumption of IoT nodes are thoroughly investigated [76] and must be adjusted based on geospatial features and the IoT node attributes. However, the correlation with the communication link is not linear, as

TABLE II: Parameters in energy consumption modelling

Parameter	Description
$\text{Energy}_{\text{Total}}$	The overall energy consumption for communicating to the sensor through the transmitter during one machine cycle
E_{active}	The energy utilization for the period of the active mode
E_{sleep}	The energy consumption during the sleeping mode by modules such as micro-controllers, transceivers, and sensors
$E_{\text{sensor\_ON}}$	The energy consumed by the sensor during the active mode
$E_{\text{micro\_ON}}$	The energy consumed by the micro-controller while processing the data
P_{sleep}	The power utilization for the period of sleeping mode
$P_{\text{sensor\_ON}}$	The power consumed by the sensor during the active mode
P_{micro}	The power consumed by the micro-controller during the active mode
P_{RX}	The power consumption during receiving mode
P_{TX}	The power consumption during transmission mode
$T_{\text{sensor\_ON}}$	The period in which a sensor is in active mode
$\text{Time}_{\text{Total}}$	The total time comprises of $\text{Time}_{\text{active}}$ and $\text{Time}_{\text{sleep}}$
$\text{Time}_{\text{active}}$	The amount of time for an IoT node in the awake state with the radio on
$\text{Time}_{\text{sleep}}$	The period just before data communication initiation in which the radio is off

multiple nodes transmitting at full power can generate interference in the network [77]. This shows that it is possible to reduce the energy consumption of the device and improve the network reliability by adjusting the transmission power of each end node. Equation 2 shows the strong dependency between the total energy consumption of an IoT node in a different set of operating modes and the associated power consumption. On the other hand, there is a strong association between the power consumed for establishing a transmission on the transmitter side and the received signal strength on the receiver part based on Equation 3. In this equation, the η parameter is highly sensitive to the propagation environment and estimated by a larger value for a scenario with a higher number of obstacles between a transmitter and a receiver. The RSSI and the power of the received signal contain other crucial communication information, such as the transmission range between IoT nodes in wireless sensor networks [78].

Consequently, all features in the dataset but *Protocol* are theoretically correlated with the energy consumption of an IoT node. This article aims to validate how effective other transmission parameters, such as *Protocol*, are against the energy modeling of IoT nodes in wireless sensor networks.

As illustrated in Fig. 1 and excluding the *Energy* feature, there is only a relatively high negative correlation between RSSI-Distance and a high positive correlation between Distance-X and Distance-Scenario among all possible cases. There is also a minimum correlation between *Protocol* and all other features. While enhancing the decision-making process, such a low correlation avoids providing similar

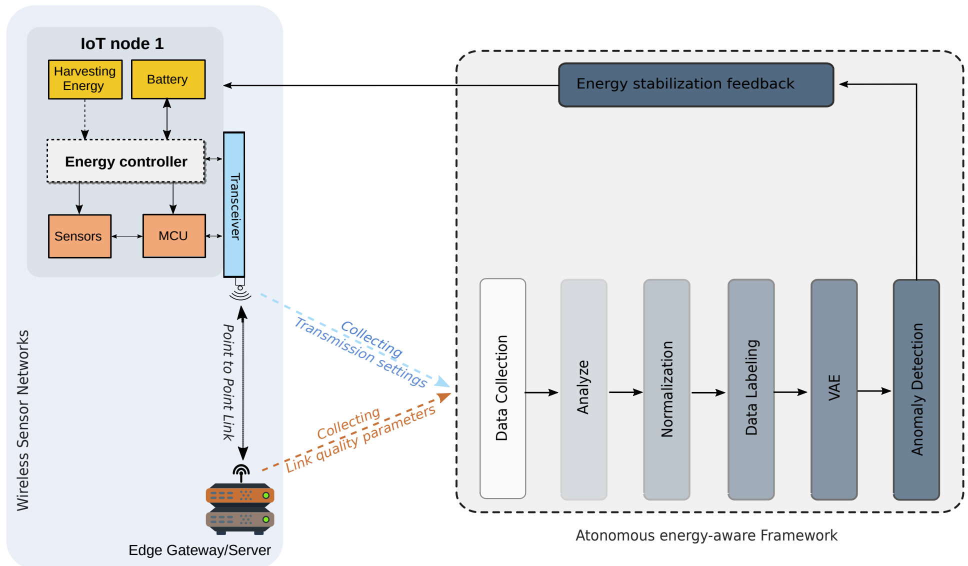


Fig. 3: The proposed energy-aware Framework

results and increasing the execution time [73].

V. Methodology

In this section, we explain our proposed machine learning-based approach for energy anomaly detection at the edge. First, we define what an anomaly is in our scenario and how a sample can disrupt a predictable pattern in the data. Next, we discuss two deep unsupervised models for energy anomaly detection and propose a deep learning-based model as a novel specialized approach to this problem. In this section, we propose an energy-aware framework for a point-to-point (P2P) scenario to design an energy-saving framework at the edge of an energy-hungry wireless sensor network (Fig. 3). The solution is based on a data analytic approach capable of detecting the over-energy-consumed IoT nodes during their transmission to the edge servers using various communication configurations. The decision-making process analyzes the observations based on the geographical location, environmental conditions, communication parameters, and link quality data. To evaluate the performance of each connection using different protocols, we study a public P2P dataset containing IoT communication energy and link quality data. Its contributors developed a script and implemented it on two ESP32 system-on-chip microcontrollers to emulate two wireless sensor endpoints. The dataset includes the X-Y coordinates of nodes obtained through GPS connection relative to the coordinate of the edge gateway as set to (0,0). To collect the RSSI value, multiple transmissions are investigated under various states, such as the position of nodes, the possibility of having indoor/outdoor scenarios, different node-gateway distances, and the number of intervening obstacles in between. By recording the communication parameters, such as transmission power consumption, and the RSSI of the signal, an array of data is created for every established link to the edge gateway. Additionally, all of the above-mentioned features are tested for the following communication protocols in data transmission: ESPNow, BLE, Radio Frequency 434MHz, LoRa, and ZigBee.

TABLE III: Protocol datasheet

Protocol	Transmission Power (dBm)	Energy Consumption (mA)
BLE	7	150
RF	20	150
LoRa	23	120
ESP-NOW	1	88
ZigBee	8	40

A. Data Labeling

The dataset used in our study comprises the communication configuration and energy consumption for P2P IoT nodes in wireless sensor networks. As the essential part of excessive energy detection lies in distinguishing normal and abnormal consumption, binary labeling fits well in this scenario. Table III describes the transmission power and the corresponding transmission energy consumption for different communication protocols, according to the datasheets. The power consumption entries in this table are the highest official values for each certain protocol in the dataset. Thus, we decided to distinguish normal and abnormal data by defining the maximum energy value as a threshold for data labeling. In other words, the anomaly threshold setting is considered as any energy use of more than 150 mA in data transmission. Any resultant energy entry provided in the dataset is designated as an anomaly if it surpasses the maximum energy consumption of all protocols (150 mA) according to the datasheets (Table III). To this end, the actual energy consumption in the dataset for different protocols is measured and provided using multiple values for all the variables, including transmission power. To detect energy anomalies in this array, we train an autoencoder exclusively on the normal energy data and later expect it to reconstruct the whole data while minimizing the reconstruction error. These observations will be classified as anomalies if the error exceeds a certain threshold defined by the maximum energy consumption of all the protocols under the experiment (Table III). It is assumed that anomalies generate much higher reconstruction errors than normal data.

B. *Feature Importance*

In training a machine learning model, each feature has a different importance that can be represented by a score. A higher score leads to a more significant impact on the model, while some have minor influences and can be ignored or removed from a dataset. In this part, to obtain each score, a basic discrete choice model is used. It is a linear regression model that is fitted to the dataset. A coefficient's absolute value (score) of a feature indicates the impact of that feature change on the energy consumption of nodes. The mentioned model is shown in Equation 4, where variables X_1 to X_8 represent features and $\hat{\text{Y}}$ represents the final output.

$$\hat{\text{Y}} = \sum_{i=1}^{8} C_i X_i \tag{4}$$

Algorithm 1 illustrates the entire process of measuring the contributions of individual input features to the performance of our learning model.

Algorithm 1 Feature scoring

Initialize a linear regression model with random coefficients: $\hat{Y} = F(X_1, \ldots, X_8)$, where $\hat{Y}$ represents the predicted values.
Fit all samples (X_i, Y_i) to the model to obtain the initial predicted values: $\hat{Y}^{(0)} = F(X_1, \ldots, X_8)$.
Calculate the residual error vector: $\epsilon^{(0)} = Y - \hat{Y}^{(0)}$, where $\epsilon^{(0)}$ represents the vector of residuals.
Set a threshold value for convergence: δ, where δ is a predefined tolerance.
while not converged (convergence is defined as $\|Y - \hat{Y}^{(k)}\| \leq \delta$) **do**
  Update the model coefficients by fitting the data again: $\hat{Y}^{(k+1)} = F(X_1, \ldots, X_8)$.
  Calculate the updated residual error vector: $\epsilon^{(k+1)} = Y - \hat{Y}^{(k+1)}$.
  if $\|Y - \hat{Y}^{(k+1)}\| \leq \delta$ **then**
    Stop the process.
  end if
end while
Sort all coefficients in descending order based on their magnitude.

Fig. 4 shows the absolute value for the coefficients of the features. As apparent, *Protocol* has the highest coefficient absolute value compared to other features and has the most significant impact on the final output. On the other hand, several features such as *X*, *Y*, *Distance*, *Power*, and *RSSI* have a lower contribution to the performance of the outcome.

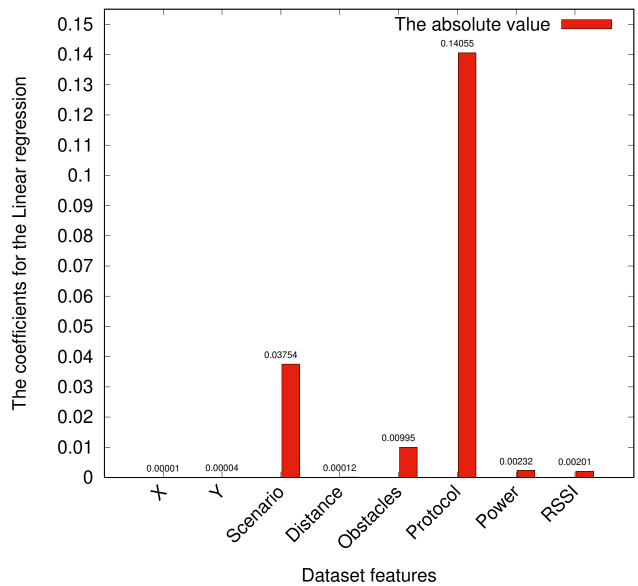


Fig. 4: The significance of the features' impact on the energy consumption of nodes

C. *Data Normalization*

The features of the utilized dataset have different distributions. This variation in the training process is typically challenging for machine learning models. We normalize each feature value to a range between zero and one to address this issue. Equation 5 illustrates the normalizing formula used in this paper. In this case, X_{min} and X_{max} correspondingly represent the minimum and maximum feature values, and X_{scaled} shows the normalized version of a feature's value.

$$\text{X}_{scaled} = (X - X_{\min})/(X_{\max} - X_{\min}) \tag{5}$$

D. *Loss Function Design*

As discussed in Section V-B, a linear regression model allowed us to sort all features based on their impacts on the final output. The *Protocol* is found to be the most significant feature among others. In this part, we want to redesign the loss function of the conventional VAE by integrating *Protocol* into its definition. The goal is to guide the direction of gradient flow to focus more on the error between the *Protocol* feature in the original and reconstructed versions of data and adjust the model parameters accordingly. As a means of accomplishing it, the loss function is redesigned by adding a new term that represents the *Protocol* value error on the input and output of the model. Equation 6 demonstrates the new loss function ($\mathcal{L}_{\text{New}}$), where the new term contains the absolute value's square of the difference between the *Protocol* feature value in the original and reconstructed versions of data. $\mathcal{L}_{\text{recon}}$ is the standard reconstruction loss (e.g., MSE or binary cross-entropy) while the $\mathcal{L}_{\text{KL}}$ is the Kullback-Leibler divergence that regularizes the learned latent space. λ is a weighting coefficient that controls the impact of the *Protocol* term in the overall loss. In this formula, β is a hyperparameter that balances the $\mathcal{L}_{\text{New}}$ and $\mathcal{L}_{\text{recon}}$. It is common to set the value for β equal to 1 in a standard VAE. Moreover, we set λ to 1 to emphasize that the relevant *Protocol* term will have the same weight as the other components of the loss function.

$$\mathcal{L}_{\text{New}} = \mathcal{L}_{\text{recon}} + \beta \cdot \mathcal{L}_{\text{KL}} + \lambda \cdot \left(\left| \hat{X}_{\text{protocol}} - X_{\text{protocol}} \right| \right)^2 \tag{6}$$

E. Deep Unsupervised Learning Model for Anomaly Detection

The proposed framework is divided into training and testing phases. First, we split the dataset into two training and testing parts. While all training samples are composed of normal observations, testing samples include both normal and abnormal data. The principal assumption is to feed normal observations into the model during training to motivate it to learn the normal data pattern. For this purpose, we apply an encoder on training samples to convert data to a latent feature space in the bottleneck layer. Then, a decoder network is applied to the latent data to reconstruct the original input. We then utilize the mean squared logarithmic error (MSLE) to define the error between the original input data and the reconstructed version. Equation 7 illustrates the MSLE loss function, where X and $\hat{X}$ represent the original and reconstructed input data, respectively. In this equation, N demonstrates the number of elements in the input data sample.

$$\text{MSLE}(X, \hat{X}) = \frac{1}{N} \sum_{i=0}^{N} \left(\log\left(X_i + 1\right) - \log\left(\hat{X}_i + 1\right) \right)^2 \tag{7}$$

Next, we define a threshold value based on the MSLE loss function that indicates the maximum error between the original and reconstructed input data. We obtain the threshold value during the training phase when we feed only normal samples to the model; therefore, the threshold indicates the maximum margin between the normal original and reconstructed input data. Equation 8 depicts the threshold formula, which consists of two terms. The first and second terms represent the mean and standard variation of the MSLE loss function, respectively, and l_i shows the ith element of the MSLE loss value.

$$\text{Threshold} = \underbrace{\frac{1}{N} \sum \text{MSLE}(X, \hat{X})}_{\text{Mean}} + \underbrace{\sqrt{\frac{\sum \left(l_i - \frac{1}{N}\text{MSLE}(X, \hat{X}) \right)^2}{N}}}_{\text{Standard deviation}} \tag{8}$$

Next, we compute the MSLE loss between the original and reconstructed test samples in the testing phase. If the MSLE loss value of each test observation is higher than the threshold, the observation is abnormal, and vice versa.

Algorithm 2 Anomaly Detection

Input: Mixed data
Output: Normal and abnormal data
Training Phase:
Feed the Encoder with normal data: $\mu, \sigma^2 = \text{Encoder}(X_{\text{train}})$
Set the latent variable Z as $Z = \mu + \epsilon \cdot e^{\sigma^2/2}$, where $\epsilon \sim \mathcal{N}(0, 1)$
Feed the Decoder with Z: $\hat{X}_{\text{train}} = \text{Decoder}(Z)$
Compute $\text{MSLE}(X_{\text{train}}, \hat{X}_{\text{train}})$ based on Equation 7
Compute a threshold value based on Equation 8
Compute ∇L by using Equation 6: $\theta, \phi \leftarrow \text{Adam}(\nabla L, \theta, \phi, \alpha)$
Testing Phase:
Feed the Encoder with sample i: $\mu_i, \sigma_i^2 = \text{Encoder}(X_{\text{test}_i})$
Set a latent variable Z_{test_i} as $Z_{\text{test}_i} = \mu_i + \epsilon \cdot e^{\sigma_i^2/2}$
Feed the Decoder with Z_{test_i}: $\hat{X}_{\text{test}_i} = \text{Decoder}(Z_{\text{test}_i})$
Compute $\text{MSLE}(X_{\text{test}_i}, \hat{X}_{\text{test}_i})$ based on Equation 7
if $\text{MSLE}(X_{\text{test}_i}, \hat{X}_{\text{test}_i}) >$ threshold **then**
    X_{test_i} is an anomaly.
else
    X_{test_i} is normal.
end if

Algorithm 2 shows the summary of the mentioned entire process in a systematic fashion. ∇L is the gradient of a loss function that measures the disparity between the predicted values generated by the model and the actual values obtained from the data. Basically, it is a vector that indicates the direction of the most rapid increase of the loss function. θ and ϕ are parameters (weights) of the model to be explored for optimal values that makes loss function minimized. The hyperparameter α is responsible for changing model parameters after each model weight update, according to the estimated error. Technically, an appropriate selection of α has a significant impact on the convergence of the training process.

VI. Experiment

In this paper, we seek to verify the performance of the energy over-consumption detection framework through the transmission settings and link quality parameters. The data is generated in a point-to-point data transmission between an IoT node and an edge gateway in a wireless sensor network. It is important to note that the design of our proposed framework is transmission-oriented. Therefore, we can simply scale up the concept to many transmissions between a node and multiple gateways.

A. Experiment Design

In this section, we describe the design of the experiment and discuss the result of the deep neural network models for detecting anomalies in the energy consumption of IoT devices. In practice, online anomaly detection frameworks could be designed as "continuously active" [79] and "active on-demand" [80]. Our proposed framework is designed to work on demand. In other words, we use this framework for diagnosis purposes. If the edge network requires energy troubleshooting, the framework starts to collect the communication arguments only for a bounded period of time. Data collection in the proposed framework follows the least intrusive technique which is called dynamic binary instrumentation. DBI (Dynamic Binary Instrumentation) [81] is known as a run-time executable modification technique that enables a tracing mechanism to collect detailed software/hardware granular data with the least perturbation [9] [82] [83] [84]. Tracing imposes an extremely low overhead and therefore it is unlikely to impact the energy consumption of nodes significantly [85] [86]. Since dynamic binary instrumentation is a run-time mechanism for collecting desired data, it does not impose any long-term overhead on the devices under experiment and the overhead of data collection could be safely lifted on the nodes immediately after the diagnosis ended. Since the troubleshooting procedure is designed to be performed in the limited time of diagnosis in our scenario, it is undeniable that the cost of our short-term analysis compensates for the cumulative overconsumption in the long run for IoT anomalous nodes.

According to equations 2 and 3, the energy consumption of an IoT node has an organic relationship with all of the parameters such as T (Transmission time), P (Transmission power), d (Transmission distance), RSSI, etc. To evaluate the possibility of having a correlation between energy consumption and other transmission parameters, a public dataset is used as a testbed for the experiment. Our proposed framework is expected to integrate the imbalanced attention for features and provide better performance metrics than traditional machine learning models and VAEs.

For the dataset, the influential features associated with the energy consumption of IoT devices are extracted by analyzing the features of experiments through a covariance matrix and linear regression. Impressively, we verify a strong *Protocol-Energy* correlation as a first step. By Finding that *Protocol* is the most correlated parameter with energy in the dataset, we accordingly design a new loss function for our neural network (variational autoencoder) by adding the error between the *Protocol* value in the original and reconstructed output to guide the direction of gradient flow. According to the binary labeling strategy for anomaly detection scenarios, multiple baseline clustering algorithms have been applied to evaluate the efficiency of those models. The given scores for these models are calculated based on their clustering estimations' accuracy. By investigating how performance metrics vary for most baseline clustering algorithms, attention is drawn to the key role of data labeling in anomaly detection. In our case, the data labeling strategy discussed in Section V-A is set based on the official datasheets of the available protocols under the experiment. Such an approach not only defines a realistic threshold for model training but also makes the final result comprehensible.

An activation function is the main part of the neural network design as it describes the transformation of the weighted sum of an input to an output from the node(s) in a network layer. We provide comparisons to demonstrate how configuration parameters are associated with the model's performance in neural networks. This section uses popular activation functions to conduct the comparison experiment. Sigmoid function as a non-linear activation function transforms the values to a range between 0 and 1. Secondly, as a symmetric activation function, the Tanh function carries out the transformation while the output range is constrained between -1 and 1. Regarding the last activation function, the ReLu, as a computationally efficient function, activates the neurons for which the output of linear transformation gets above zero and deactivates them otherwise.

In addition to the provided performance metrics, we evaluated the indirect complexity of the proposed algorithm by measuring the model execution time. It contains the training and testing time of the model that relatively represents how feasible the model implementation is considering other constraints imposed in certain scenarios.

B. Result and Discussion

By modifying and training a 2-layer VAE with normal energy data and testing it on a mix of normal and abnormal energy consumption observations, the performance metrics and execution time are compared to a classic 2-layer VAE in Fig. 5. This figure depicts the performance of a 2-layer VAE against the proposed 2-layer modified VAE model measured by accuracy, precision score, F1 score, MAE, RMSE, training time, and testing time using three active functions. The performance metrics and execution time (training/testing time) computation are designed to get the advantage of multiple K from 1 to 5 for a K-fold setting.

With Tanh as an activation function, the proposed 2-layer modified variational autoencoder performs best (Fig. 5a-5e). While the performance of 2-layer VAEs for ReLu and Sigmoid degrades compared to the proposed modified version, the anomaly detection capability of 2-layer VAE for Tanh improves in its modified form. Meanwhile, none of the proposed 2-layer modified VAEs could gain any better performance than that of the 2-layer VAE in accuracy, precision score, F1 score, MAE, RMSE levels, and execution time. Based on Fig. 5f and 5g, the execution time for each activation function increases in the modified form. It makes sense as the newly designed loss function has an additive term that comes with additional time for processing and computation (see Equation 6). These figures not only confirm that none of the performance metrics for the proposed modified 2-layer VAE improved but also indicate an 11 and 5 percent increase in training and testing time (ReLu) compared to that of the 2-layer VAE (ReLu). Besides, the ReLu activation function takes the least training and testing time of all. Therefore, considering all the activation functions, the 2-layer VAE using the ReLu

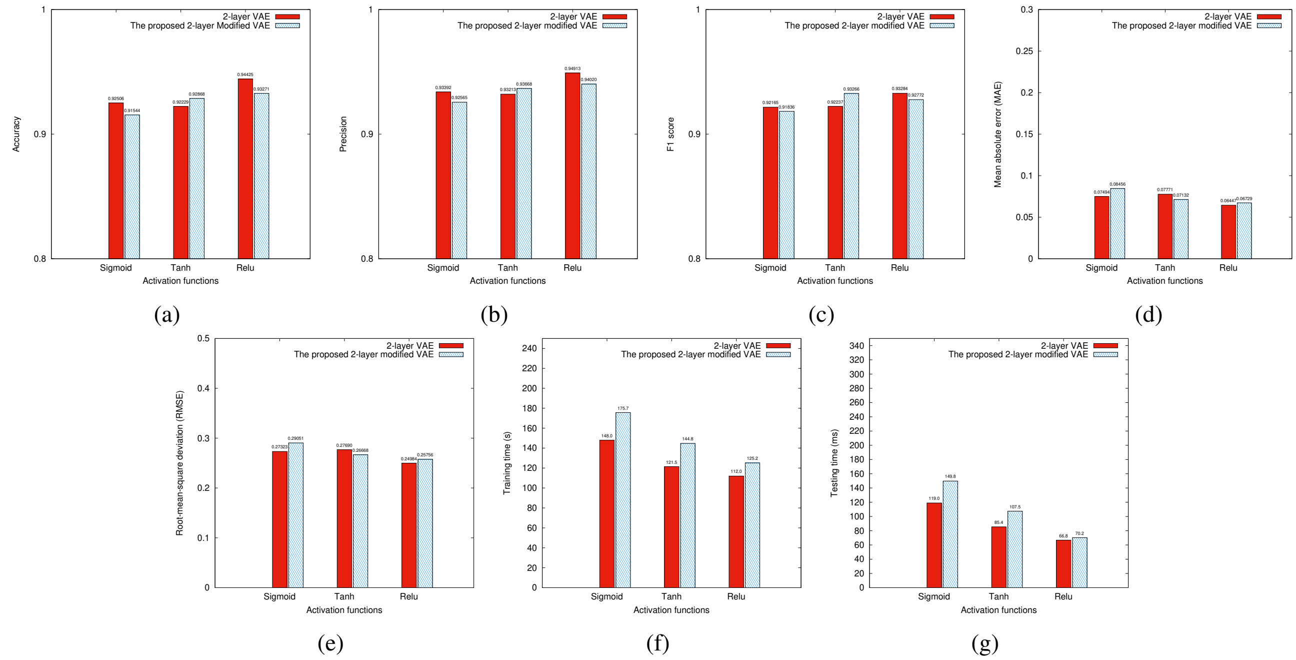


Fig. 5: A comparison between performance metrics and execution time for 2-layer VAE and the proposed 2-layer modified VAE using different activation functions: (a) Accuracy; (b) Precision score; (c) F1 score; (d) MAE; (e) RMSE; (f) Training time; and (g) Testing time

activation function outperforms the other configurations. To better evaluate the model performance, a different number of hidden layers is inserted as the depth of our neural network. Fig. 6 represents performance metrics, execution time, and configuration for a 3-layer VAE and the proposed 3-layer modified VAE to investigate the effect of the number of hidden layers on the efficiency of our model. By comparing the performance among the three activation functions for different K parameters, the proposed 3-layer modified VAE using the ReLu activation function accomplished the best. It achieved a higher accuracy, precision score, and F1 score while obtaining lower MAE and RMSE values at the cost of 12 and 2 percent increase for the training and testing time, respectively. Fig. 6f and 6g illustrate how temporally efficient the ReLu activation function is compared to other counterparts both in the modified and unmodified 3-layer VAE. One of the advantages of ReLu in obtaining the best execution time lies in sparsity, as a majority of the neurons are inactivated in a layer for a given input. The fewer number of neurons staying active, the lower the computational load and time would be. By keeping all the neurons active in the other activation functions, Sigmoid and Tanh are computationally expensive. In the case of Sigmoid, the execution time is clearly larger than that of others. This activation function contains an exponential term in its definition. This term requires a series expansion and a complex numerical approximation that makes its computation costly, in particular for large values of input. Sigmoid also involves a division operation which adds extra overhead to the computational cost. Moreover, the symmetry in the Sigmoid function requires the computation of both negative and positive values of the input, thus doubling the computational cost compared to non-symmetric activation functions. At this point, comparing the performance metrics and execution time of a 2-layer VAE using ReLu against the proposed 3-layer modified VAE using ReLu activation function results in a comprehensive analysis of our investigation. Considering the three activation functions, Fig. 7 shows that the highest accuracy, precision, and F1 score in parallel with the lowest MAE and RMSE belong to the proposed 3-layer modified VAE. The highest performance metrics level is obtained with three hidden layers for the ReLu activation function at the cost of 14 and 17 percent increase for the training and testing time, respectively. As already discussed, our diagnostic framework performs in an ”active on-demand” mode for a short duration in practice. Therefore, such overhead for a troubleshooting framework can be tolerated by the network compared to the driven benefits of conserving excessive amounts of energy consumption in a lifetime of anomalous nodes.

Although 1 or 2 percentages of improvement are negligible for performance metrics with values lower than 90% level, it becomes remarkably challenging to make further improvements as a model becomes more accurate than 90%. In other words, as the 2-layer VAE has already obtained high performance (above 90%), even small gains for the proposed 3-layer modified VAE can be significantly important to achieve because we are operating in a region where errors are inherently hard to reduce.

A comprehensive overview of the performance metrics of

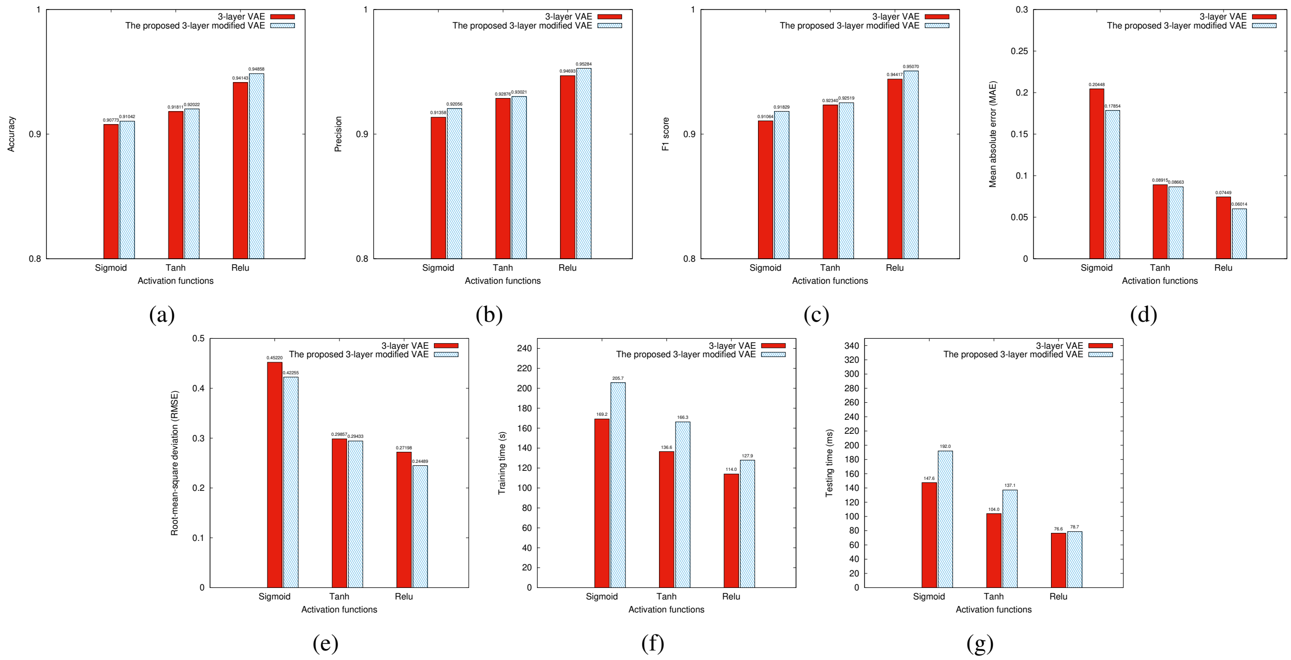


Fig. 6: A comparison between performance metrics and execution time for 3-layer VAE and the proposed 3-layer modified VAE using different activation functions: (a) Accuracy; (b) Precision score; (c) F1 score; (d) MAE; (e) RMSE; (f) Training time; and (g) Testing time

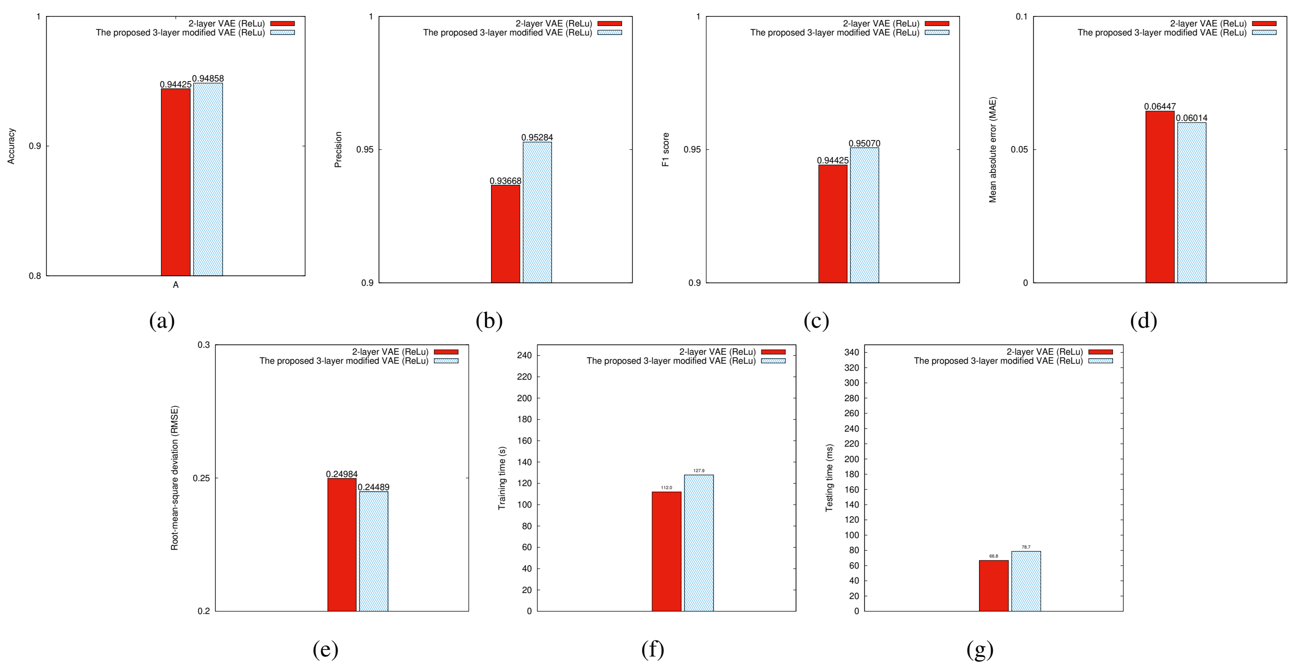


Fig. 7: A comparison between performance metrics for 2-layer VAE and the proposed 3-layer modified VAE using ReLu activation function: (a) Accuracy; (b) Precision score; (c) F1 score; (d) MAE; (e) RMSE; (f) Training time; and (g) Testing time

TABLE IV: Comparison between the baseline clustering algorithms and the proposed modified VAE

Models	Accuracy	Precision Score	F1 Score
Kmeans	0.706	0.857	0.774
Normalized Kmeans	0.640	0.829	0.722
Mean Shift	0.594	0.858	0.702
Normalized Mean Shift	0.646	0.847	0.733
Spectral Clustering	0.552	0.872	0.676
Normalized Spectral Clustering	0.569	0.845	0.680
Birch	0.621	0.857	0.720
Normalized Birch	0.640	0.829	0.722
DBSCAN	0.150	0.860	0.260
Normalized DBSCAN	0.920	0.840	0.880
Mixture Model	0.713	0.857	0.778
Normalized Mixture Model	0.644	0.848	0.732
BGMM	0.477	0.857	0.613
Normalized BGMM	0.640	0.829	0.722
Agglomerative Clustering	0.621	0.857	0.720
Normalized Agglomerative Clustering	0.640	0.829	0.722
Normalized 3-layer Modified VAE using ReLu activation function	0.948	0.952	0.950

the several baseline clustering models and the proposed 3-layer modified VAE is presented in Table IV. According to the comparison, the performance of our proposed 3-layer modified VAE is improved dramatically not only for accuracy but also for precision score and F1 score against the baseline models. To evaluate the efficiency of the normalized 3-layer modified VAE model, we concentrate on the performance metrics in Table IV and Fig. 7. The output of Sigmoid and Tanh activation functions is saturated for large positive and negative inputs. Such poor performance results in the vanishing gradient problem. This is a common problem in creating neural networks [87]. It originates from flattening the derivative of the activation function. Therefore, there will be almost no new updates on parameters in the process of model training which leads to poor convergence. Similar to the pruning mechanisms, the Rectified Linear Unit (ReLu) can invalidate the statistically nonessential features by putting them to zero. In other words, the linear structure of ReLu (non-saturating feature) boosts model training efficiency as the output of the partial derivative of the loss function will be mapped to a binary result (0 or 1) and consequently stops the gradient from vanishing.

We find out that identifying the dominant features and redesigning the loss function accordingly, gives the network a deeper understanding of distinguishing the normal and abnormal data patterns. Our investigation manifests that choosing the proposed 3-layer modified VAE using the ReLu activation function improves the robustness of the model according to the obtained values for accuracy, precision score, F1 score, MAE, RMSE, training time, and testing time.

Eventually, our investigation proves that unfit adjustment of link parameters in data transmission, such as the *Protocol* plays a key indicator of high energy consumption between an IoT device and the edge elements. Based on the protocol type, intuitive assumptions could be devised for the reason behind the overconsumption of resource-constrained devices. For security protocols, those which require significant cryptographic processing with costly cipher parameters, increase the energy consumed by the battery-powered nodes. For communication protocols, those that are susceptible to collisions (large packet loss) made by simultaneous flooding of tag responses are energy-exhaustive mechanisms. For routing protocols, those determining routes statically and relying on periodic advertisements generated by routers consume significantly more energy than traditional ones. It should be declared that the online frameworks equipped with energy-aware protocols should also take care of a limited time window for service provisioning without compromising QoS in order to avoid operation interruption.

According to the provided statistics, we can verify that the delivered anomaly detection framework could be appointed as a validated approach for energy over-consumption analysis. Further, the energy stabilization feedback enables us to alter the transmission settings which determine the alternative with the least energy footprint, under any given operating environment. It may toggle the node configuration towards a greener setting. In such settings, routing protocols, security protocols, and communication protocols will meet energy-awareness standards, while network responsiveness ensures the necessary QoS levels and reliability.

VII. Conclusion

Artificial-intelligence-integrated Internet of Things (AIoT) is an emerging discipline that integrates AI analysis and approaches into the Internet of Things field of interest. By nature, applications running in the AIoT domain are computation-intensive. They require a high level of real-time processing to achieve decisions made by ML (Machine Learning), DL (Deep learning), and data analysis operations. The high number of constraints imposed on an AIoT scenario, such as limited energy harvesting potential and demanding energy consumption, makes achieving a better quality of service (QoS) challenging for applications. Since ignoring the energy anomalies can easily result in battery depletion and network failure eventually, our work focuses on the early detection of the hotspots communicating with the edge gateway according to their link quality and communication configurations. Accordingly, this paper proposes a deep learning approach and a binary labeling strategy for identifying energy-intensive operations. Such a self-configuring scheme uses a 3-layer modified VAE to detect anomalies for peer-to-peer connections using link quality and communication features. Additionally, several unsupervised clustering algorithms are also presented, validated, and compared with the performance metrics of our model, including the Mixture Model, Bayesian Gaussian Mixture Model, Agglomerative Clustering, Birch, and Modified Variational Autoencoder. By analyzing these performance metrics, the best efficiency in anomaly detection for the energy consumption of IoT nodes in wireless sensor networks is achieved. As shown earlier, the proposed 3-layer modified VAE model reached an outstanding performance of 94.8% accuracy, 95.2% precision score, 95.0% F1 score, 0.06014 MAE, and 0.24489 RMSE for anomalous energy detection.

As a primary conclusion, we find out that an improper configuration of IoT nodes (e.g., communication protocol selection for data transmission) may considerably increase the energy consumption of nodes. Such settings subsequently stop the necessary services from executing satisfactorily and fulfilling expectations. Establishing energy-sustainable networks on edge requires a real-time monitoring system to discover the energy-hungry nodes. Most existing AI-based solutions propose offline frameworks using supervised or intrusive approaches. The supervised solutions on low-power IoT nodes are mostly invalid due to the infeasible energy labeling in real-time. High-intrusive approaches modify the native software or hardware to monitor the energy behavior of different components of the nodes. While energy sampling methods can capture the energy peaks in predetermined periods, they cannot target contextual anomalies. The major reason for such deep suffering lies in insufficient precision for complex sensing activities/computation. Moreover, providing thousands of sensors with external energy monitoring hardware is not practical, especially if flexibility, price, and physical size come into the scene. Hence, for early energy anomaly analysis and transmission parameters adjustment, we employ a deep learning approach to detect abnormal energy data indirectly without supervision.

For future work, using time-series data for transmission parameters and link quality data not only helps us identify patterns but also creates the opportunity to detect contextual anomalies and predict future patterns. Using feature interaction learning, we can also capture predictive information (interactions between features) to improve the modified loss function. Moreover, besides having the external data regarding data communication, collecting the internal data such as CPU/GPU/RAM utilization and NIC I/O rate gives us more data to train the model. We hypothesize that such a model not only detects the anomalous nodes in an underconsumption state but can minimize the number of false positive anomalies as it is contextually bound to the internal behavior of the system rather than just a constant threshold setting.

VIII. ACKNOWLEDGMENTS

We would like to thank the Humanitas Corporation and MITACS (The Mathematics of Information Technology and Complex Systems) organization for supporting this research.

REFERENCES

[1] X. Li, J. Peng, J. Niu, F. Wu, J. Liao, K.-K. R. Choo, A robust and energy efficient authentication protocol for industrial internet of things, IEEE Internet of Things Journal (2018). doi:10.1109/JIOT.2017.2787800.

[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, Industrial internet of things: Challenges, opportunities, and directions, IEEE Transactions on Industrial Informatics (2018). doi:10.1109/TII.2018.2852491.

[3] L. Dash, M. Khuntia, Energy efficient techniques for 5g mobile networks in wsn: A survey, 2020. doi:10.1109/ICCSEA49143.2020.9132941.

[4] E. S. Ali, R. A. Saeed, I. K. Eltahir, O. O. Khalifa, A systematic review on energy efficiency in the internet of underwater things (iout): Recent approaches and research gaps, Journal of Network and Computer Applications (2023). doi:https://doi.org/10.1016/j.jnca.2023.103594.

[5] Y. Sun, J. Wang, X. Wang, Fault diagnosis of mechanical equipment in high energy consumption industries in china: A review, Mechanical Systems and Signal Processing 186 (2023). doi:https://doi.org/10.1016/j.ymssp.2022.109833.

[6] Y. Padarthi, R. R. P. Kuppusamy, Iot-based embedded sensor system for real-time health monitoring of composite structures for large-scale industrial operations, CRC Press, 2023.

[7] D. Zhai, C. Wang, R. Zhang, H. Cao, F. R. Yu, Energy-saving deployment optimization and resource management for uav-assisted wireless sensor networks with noma, IEEE Transactions on Vehicular Technology (2022). doi:10.1109/TVT.2022.3159681.

[8] C. Niemann, C. Ewert, H. Puttnies, M. Rethfeldt, D. Timmermann, P. Danielis, Modeling energy consumption for task-offloading decisions on mobile and embedded devices, 2020. doi:10.1109/LifeTech48969.2020.1570618809.

[9] A. S. Bushehri, S. Keivanpour, M. Azam, G. Niculescu, Anomalous energy detection for resource-constrained embedded systems using tracing data analysis, 2021. doi:10.1109/ICECET52533.2021.9698794.

[10] S. Dey, A. K. Singh, S. Saha, X. Wang, K. D. McDonald-Maier, Rewardprofiler: A reward based design space profiler on dvfs enabled mpsocs, 2019. doi:10.1109/CSCloud/EdgeCom.2019.00028.

[11] A. Katal, S. Dahiya, T. Choudhury, Energy efficiency in cloud computing data centers: a survey on software technologies, Cluster Computing (2022).

[12] P. Bratek, L. Szustak, R. Wyrzykowski, T. Olas, Reducing energy consumption using heterogeneous voltage frequency scaling of data-parallel applications for multicore systems, Journal of Parallel and Distributed Computing (2023). doi:https://doi.org/10.1016/j.jpdc.2023.01.005.

[13] B. Peccerillo, M. Mannino, A. Mondelli, S. Bartolini, A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives, Journal of Systems Architecture (2022). doi:https://doi.org/10.1016/j.sysarc.2022.102561.

[14] P. Dibal, E. Onwuka, S. Zubair, E. Nwankwo, S. Okoh, B. A. Salihu, H. Mustaphab, Processor power and energy consumption estimation techniques in iot applications: A review, Internet of Things (2023). doi:https://doi.org/10.1016/j.iot.2022.100655.

[15] J. Chen, M. Manivannan, M. Abduljabbar, M. Pericàs, Erase: Energy efficient task mapping and resource management for work stealing runtimes (2022). doi:10.1145/3510422.

[16] A. Iranfar, M. Kamal, A. Afzali-Kusha, M. Pedram, D. Atienza, Thespot: Thermal stress-aware power and temperature management for multiprocessor systems-on-chip, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2018). doi:10.1109/TCAD.2017.2768417.

[17] P. Mishra, N. Kumar, W. W. Godfrey, An evolutionary computing-based energy-efficient solution for iot-enabled software-defined sensor network architecture, International Journal of Communication Systems (2022). doi:https://doi.org/10.1002/dac.5111.

[18] J. Toledo-Castro, P. Caballero-Gil, N. Rodríguez-Pérez, I. Santos-González, C. Hernández-Goya, R. Aguasca-Colomo, Forest fire prevention, detection, and fighting based on fuzzy logic and wireless sensor networks, Complexity (2018).

[19] R. Pachlor, D. Shrimankar, Vch-eccr: A centralized routing protocol for wireless sensor networks, Journal of Sensors (2017).

[20] G. M. Abdulsahib, O. I. Khalaf, Accurate and effective data collection with minimum energy path selection in wireless sensor networks using mobile sinks, Journal of Information Technology Management (2021).

[21] D. E. Ruíz-Guirola, C. A. Rodríguez-López, S. Montejo-Sánchez, R. D. Souza, M. A. Imran, Drx-based energy-efficient supervised machine learning algorithm for mobile communication networks, IET Communications (2021). doi:https://doi.org/10.1049/cmu2.12137.

[22] O. Khalaf, C. Romero, S. Hassan, M. T. Iqbal, Mitigating hotspot issues in heterogeneous wireless sensor networks, Journal of Sensors (2022). doi:10.1155/2022/7909472.

[23] B. A. Begum, S. V. Nandury, Data aggregation protocols for wsn and iot applications – a comprehensive survey, Journal of King Saud University - Computer and Information Sciences (2023). doi:https://doi.org/10.1016/j.jksuci.2023.01.008.

[24] C. Jiang, Y. Qiu, H. Gao, T. Fan, K. Li, J. Wan, An edge computing platform for intelligent operational monitoring in internet data centers, IEEE Access (2019).

[25] E. Fazel, H. E. Najafabadi, M. Rezaei, H. Leung, Unlocking the power of mist computing through clustering techniques in iot networks, Internet of Things (2023). doi:https://doi.org/10.1016/j.iot.2023.100710.

[26] O. Kanoun, S. Bradai, S. Khriji, G. Bouattour, D. El Houssaini, M. Ben Ammar, S. Naifar, A. Bouhamed, F. Derbel, C. Viehweger, Energy-aware system design for autonomous wireless sensor nodes: A comprehensive review, Sensors (2021).

[27] T. Rault, A. Bouabdallah, Y. Challal, Energy efficiency in wireless sensor networks: A top-down survey, Computer networks (2014).

[28] P. Jayakrishna, T. Sudha, Energy efficient wireless sensor network assisted spectrum sensing for cognitive radio network, IEEE, 2017.

[29] G. P. Joshi, S. Y. Nam, S. W. Kim, Cognitive radio wireless sensor networks: applications, challenges and research trends, Sensors (2013).

[30] M. A. Alamri, An efficient cooperative technique for power-constrained multiuser wireless network, Telecommunication Systems (2018).

[31] T. Cevik, A. Gunagwera, A directional multicasting-based architecture for wireless sensor networks, International Journal of Electronics (2019).

[32] C. Lee, J. Oh, W. Na, J. Yoon, W. Noh, S. Cho, Directional-antenna-based spatial and energy-efficient semi-distributed spectrum sensing in cognitive internet-of-things networks, Journal of Network and Computer Applications (2023).

[33] M. Abo-Zahhad, M. Farrag, A. Ali, Optimization of transmitted power and modulation level for minimizing energy consumption in wireless sensor networks, Wireless Personal Communications (2017).

[34] H. A. B. Salameh, M. Bani Irshaid, A. Al Ajlouni, M. Aloqaily, Energy-efficient cross-layer spectrum sharing in cr green iot networks, IEEE Transactions on Green Communications and Networking (2021). doi:10.1109/TGCN.2021.3076695.

[35] X. Wang, C. Zhai, Dynamic power control for cell-free industrial internet of things with random data arrivals, IEEE Transactions on Industrial Informatics (2022). doi:10.1109/TII.2021.3077892.

[36] K.-H. Hsia, C.-W. Hung, H. T. Chang, Y.-H. Lai, Transmission power control for wireless sensor network., J. Robotics Netw. Artif. Life (2017).

[37] X. Liu, M. Zhao, A. Liu, K. K. L. Wong, Adjusting forwarder nodes and duty cycle using packet aggregation routing for body sensor networks, Information Fusion (2020). doi:https://doi.org/10.1016/j.inffus.2019.06.020.

[38] O. Kanoun, Energy Harvesting for Wireless Sensor Networks: Technology, Components and System Design, Walter de Gruyter GmbH & Co KG, 2018.

[39] A. A. Shabaneh, A. M. Ali, C. K. Ng, N. K. Noordin, A. Sali, M. Yaacob, et al., Review of energy conservation using duty cycling schemes for ieee 802.15. 4 wireless sensor network (wsn), Wireless personal communications (2014).

[40] Y. Du, J. Xia, J. Gong, X. Hu, An energy-efficient and fault-tolerant topology control game algorithm for wireless sensor network, Electronics (2019).

[41] R. Kavra, A. Gupta, S. Kansal, Systematic study of topology control methods and routing techniques in wireless sensor networks, Peer-to-Peer Networking and Applications (2022).

[42] M. Demirci, A survey of machine learning applications for energy-efficient resource management in cloud computing environments, in: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015.

[43] N. Vasić, D. Novaković, S. Miučin, D. Kostić, R. Bianchini, Dejavu: Accelerating resource allocation in virtualized environments, Association for Computing Machinery, 2012.

[44] T. V. T. Duy, Y. Sato, Y. Inoguchi, Performance evaluation of a green scheduling algorithm for energy savings in cloud computing, in: 2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010.

[45] J. L. Berral, I. n. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, J. Torres, Towards energy-aware scheduling in data centers using machine learning, Association for Computing Machinery, 2010.

[46] M. Dabbagh, B. Hamdaoui, M. Guizani, A. Rayes, Energy-efficient cloud resource management, in: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2014.

[47] I. AlQerm, B. Shihada, Enhanced machine learning scheme for energy efficient resource allocation in 5g heterogeneous cloud radio access networks, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017.

[48] S. Jiang, S. R. Priya, N. Elango, J. Clay, R. Sridhar, An energy efficient in-memory computing machine learning classifier scheme, 2019.

[49] A. Sayed, A. Alsalemi, Y. Himeur, F. Bensaali, A. Amira, Endorsing energy efficiency through accurate appliance-level power monitoring, automation and data visualization, Springer, 2021.

[50] A. Alsalemi, Y. Himeur, F. Bensaali, A. Amira, An innovative edge-based internet of energy solution for promoting energy saving in buildings, Sustainable Cities and Society (2022). doi:https://doi.org/10.1016/j.scs.2021.103571.

[51] A. Alsalemi, Y. Himeur, F. Bensaali, A. Amira, Appliance-level monitoring with micro-moment smart plugs, 2021.

[52] L. Lei, B. Wu, X. Fang, L. Chen, H. Wu, W. Liu, A dynamic anomaly detection method of building energy consumption based on data mining technology, Energy (2023).

[53] A. Copiaco, Y. Himeur, A. Amira, W. Mansoor, F. Fadli, S. Atalla, S. S. Sohail, An innovative deep anomaly detection of building energy consumption using energy time-series images, Engineering Applications of Artificial Intelligence (2023).

[54] J. Kong, W. Jiang, Q. Tian, M. Jiang, T. Liu, Anomaly detection based on joint spatio-temporal learning for building electricity consumption, Applied Energy (2023).

[55] E. Shamsa, A. Kanduri, P. Liljeberg, A. M. Rahmani, Concurrent application bias scheduling for energy efficiency of heterogeneous multi-core platforms, IEEE Transactions on Computers (2021).

[56] E. Shamsa, A. Kanduri, A. M. Rahmani, P. Liljeberg, Energy-performance co-management of mixed-sensitivity workloads on heterogeneous multi-core systems, 2021.

[57] H. Ahmadvand, F. Foroutan, M. Fathy, Dv-dvfs: merging data variety and dvfs technique to manage the energy consumption of big data processing, Journal of Big Data (2021).

[58] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, R. Bianchini, Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms, 2017.

[59] S. Malik, M. Tahir, M. Sardaraz, A. Alourani, A resource utilization prediction model for cloud data centers using evolutionary algorithms and machine learning techniques, Applied Sciences (2022).

[60] J. Kumar, D. Saxena, A. K. Singh, A. Mohan, Biphase adaptive learning-based neural network model for cloud datacenter workload forecasting, Soft Computing (2020).

[61] D. Saxena, A. K. Singh, A proactive autoscaling and energy-efficient vm allocation framework using online multi-resource neural network for cloud data center, Neurocomputing (2021).

[62] M. Injadat, F. Salo, A. B. Nassif, A. Essex, A. Shami, Bayesian optimization with machine learning algorithms towards anomaly detection, 2018. doi:10.1109/GLOCOM.2018.8647714.

[63] A. Belkhiri, A. Shahnejat Bushehri, F. Gohring de Magalhaes, G. Nicolescu, Transparent trace annotation for performance debugging in microservice-oriented systems (work in progress paper), 2023.

[64] A. Belkhiri, A. S. Bushehri, M. Dagenais, Virtual networks link-layer topologies discovery through host-based tracing, IEEE, 2022.

[65] A. Belkhiri, M. Pepin, M. Bly, M. Dagenais, Performance analysis of dpdk-based applications through tracing, Journal of Parallel and Distributed Computing (2023). doi:https://doi.org/10.1016/j.jpdc.2022.10.012.

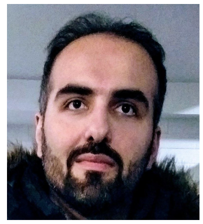
[66] A. Belkhiri, M. Dagenais, Diagnostic and troubleshooting of openflow-enabled switches using kernel and userspace traces, International Journal of Communication Systems (2021). doi:https://doi.org/10.1002/dac.4920.

[67] A. Sgueglia, A. Di Sorbo, C. A. Visaggio, G. Canfora, A systematic literature review of iot time series anomaly detection solutions, Future Generation Computer Systems (2022). doi:https://doi.org/10.1016/j.future.2022.04.005.

[68] A. Malki, E.-S. Atlam, I. Gad, Machine learning approach of detecting anomalies and forecasting time-series of iot devices, Alexandria Engineering Journal (2022). doi:https://doi.org/10.1016/j.aej.2022.02.038.

[69] M. Bor, U. Roedig, Lora transmission parameter selection, in: 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2017.
[70] A. F. X. Glória, P. J. A. Sebastião, Autonomous configuration of communication systems for iot smart nodes supported by machine learning, IEEE Access (2021). doi:10.1109/ACCESS.2021.3081794.
[71] A. F. X. d. Glória, Sustainable modular iot solution for smart cities applications supported by machine learning algorithms (2021).
[72] B. C. D. Dias, Heterogeneous communication scheme for iot smart nodes, Ph.D. thesis (2021).
[73] H. K. Yugank, R. Sharma, S. H. Gupta, An approach to analyse energy consumption of an iot system, International Journal of Information Technology (2022).
[74] Large scale survey for radio propagation in developing machine learning model for path losses in communication systems, Scientific African (2023). doi:https://doi.org/10.1016/j.sciaf.2023.e01550.
[75] W. Njima, M. Chafii, A. Nimr, G. Fettweis, Convolutional neural networks based denoising for indoor localization, 2021. doi:10.1109/VTC2021-Spring51267.2021.9448839.
[76] A. Gloria, C. Dionisio, G. Simões, P. Sebastião, Lora transmission power self configuration for low power end devices, 2019. doi:10.1109/WPMC48795.2019.9096197.
[77] R. Macharia, K. Lang'at, P. Kihato, Interference management upon collaborative beamforming in a wireless sensor network monitoring system featuring multiple unmanned aerial vehicles, 2021. doi:10.1109/AFRICON51333.2021.9570907.
[78] Z. El Khaled, W. Ajib, H. Mcheick, Log distance path loss model: Application and improvement for sub 5 ghz rural fixed wireless networks, IEEE Access (2022). doi:10.1109/ACCESS.2022.3166895.
[79] Y. Maleh, A. Ezzati, Y. Qasmaoui, M. Mbida, A global hybrid intrusion detection system for wireless sensor networks, Procedia Computer Science (2015). doi:https://doi.org/10.1016/j.procs.2015.05.108.
[80] A. Yahyaoui, T. Abdellatif, R. Attia, Read: Reliable event and anomaly detection system in wireless sensor networks, 2018. doi:10.1109/WETICE.2018.00044.
[81] C. Harper-Cyr, M. R. Dagenais, A. S. Bushehri, Fast and flexible tracepoints in x86, Software: Practice and Experience (2019).
[82] M. Schaarschmidt, M. Uelschen, E. Pulvermüller, Power consumption estimation in model driven software development for embedded systems., 2021.
[83] M. Schaarschmidt, M. Uelschen, E. Pulvermüller, Towards power consumption optimization for embedded systems from a model-driven software development perspective, Springer International Publishing, 2022.
[84] Y. B. Atitallah, J. Mottin, N. Hili, T. Ducroux, G. Godet-Bar, A power consumption estimation approach for embedded software design using trace analysis, 2015. doi:10.1109/SEAA.2015.34.
[85] M. Bourdon, P.-F. Gimenez, E. Alata, M. Kaaniche, V. Migliore, V. Nicomette, Y. Laarouchi, Hardware-performance-counters-based anomaly detection in massively deployed smart industrial devices, 2020. doi:10.1109/NCA51143.2020.9306726.
[86] R. Gassais, N. Ezzati-Jivan, J. M. Fernandez, D. Aloise, M. R. Dagenais, Multi-level host-based intrusion detection system for internet of things, Journal of Cloud Computing (2020).
[87] S. R. Dubey, S. K. Singh, B. B. Chaudhuri, Activation functions in deep learning: A comprehensive survey and benchmark, Neurocomputing (2022). doi:https://doi.org/10.1016/j.neucom.2022.06.111.

Ahmad Shahnejat Bushehri completed his Master of Science degree in Telematics Engineering at the Politecnico di Torino. Subsequently, he began his doctoral studies at Polytechnique Montreal, where he joined the Heterogeneous Embedded Systems lab and Poly Circle X.0 Lab. His Ph.D. research focuses on dynamic binary instrumentation, system tracing, energy optimization, and anomaly detection in computer systems.



Ashkan Amirnia is currently pursuing a Ph.D. degree with the Department of Mathematical and Industrial Engineering at Polytechnique Montréal, Canada. He conducts research on real-time decision-making models based on machine learning and deep learning algorithms, focusing on industrial issues, particularly in robotic/cobotic manufacturing. He is also a member of the Poly Circle X.0 lab, CIRODD (Interdisciplinary Research Centre on Sustainable Development Operationalization), and CIRRELT (Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation).

Adel Belkhiri received his Ph.D. in computer engineering from Polytechnique Montreal in 2021. Currently, he is a postdoctoral researcher at the Heterogeneous Embedded Systems Lab. His main research interests are software support for improving parallel applications performance, fault diagnosis, and observability of computer networks and distributed systems.

Samira Keivanpour is an assistant professor of mathematical and industrial engineering at Polytechnique Montréal, Canada. Her research focuses on sustainable supply chain and logistics management, especially for end-of-life products, circular manufacturing, and Industry 4.0 technologies. She is also a member of CIRODD (Interdisciplinary Research Centre on Sustainable Development Operationalization), GERAD (Group for Research in Decision Analysis) and CIRRELT (Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation).

Felipe G. de Magalhaes has a double PhD degree on Computer Engineering at Ecole Polytechnique and Computer Science at PUCRS, obtained in 2017. Currently, he is an associate researcher at Ecole Polytechnique de Montreal. Felipe has authored and co-authored around 40 papers and articles and 2 book chapters. His research interests are integrated silicon photonics, real-time systems, avionics systems, cybersecurity, implementation of multi-processor architectures and intra-chip communication infrastructure.

Gabriela Nicolescu is a professor and director of the Department of Computer and Software Engineering at Polytechnique Montréal. Her research interests lie in the field of design methods for secure and efficient architectures of current and future embedded systems. She is highly active in system-level design methods, modeling, simulation, and profiling, as well as in security by design. She collaborates with several industrial partners and academics and has published more than 240 papers in international conferences and journals.