ISSN: 2182-2069 / E-ISSN: 2182-2077

Tiny ML-Enabled Energy-Efficient Intrusion Detection System for Sustainable IoT Security in Green Cybersecurity Ecosystems

Rajalakshmi Selvaraj[1*], Venu Madhav kuthadi[2], Dr.S. Baskar[3], and Roberto Acevedo[4]

[1*]Department of Computing and Informatics, School of Pure and Applied Sciences, Botswana International University of Science and Technology, Botswana. selvarajr@biust.ac.bw. https://orcid.org/0000-0002-7059-1702

[2]Department of Computing and Informatics, School of Pure and Applied Sciences, Botswana International University of Science and Technology, Botswana. kuthadiv@biust.ac.bw. https://orcid.org/0000-0003-4515-1921

[3]Assistant Professor, Department of Electronics and Communication Engineering, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India. connectbaskar@gmail.com, https://orcid.org/0000-0003-3570-3059

[4]Facultad de Ingeniería, Universidad San Sebastian, Bellavista, Santiago. chile.roberto.acevedo.llanos@gmail.com, https://orcid.org/0000-0001-6847-0285

Received: May 28, 2025; Revised: July 12, 2025; Accepted: August 13, 2025; Published: August 30, 2025

Abstract

Internet of Things (IoT) devices face increasing security threats while operating under severe energy constraints. This research addresses the critical challenge of implementing effective cybersecurity measures within sustainable energy frameworks for resource-constrained IoT environments. The research presents Sustainable Micro-Neural Energy-Efficient Security Intelligence (SMEESI), a novel TinyML-enabled intrusion detection system (IDS) that significantly reduces energy consumption while maintaining robust security capabilities. The research innovation combines lightweight neural network architectures with energy-aware anomaly detection algorithms specifically optimized for microcontroller deployment. The SMEESI framework includes an adaptive power management module that dynamically adjusts computational intensity based on threat levels, achieving energy efficiency without compromising security posture. Performance evaluation demonstrates a 78% reduction in power consumption compared to traditional IDS implementations while maintaining 94.3% detection accuracy across multiple attack vectors. Memory footprint requirements decreased by 65%, enabling deployment on severely resource-limited IoT sensors and actuators. The system has been successfully tested in smart buildings, industrial monitoring, and healthcare IoT applications, proving its versatility across critical infrastructure domains. This research contributes to green cybersecurity by enabling sustainable security monitoring in IoT ecosystems, extending device battery life, reducing electronic waste through prolonged hardware lifecycles, and minimizing the carbon footprint of security operations while maintaining essential protection against evolving cyber threats in the digital age.

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 3 (August), pp. 602-625. DOI: *10.58346/JISIS.2025.I3.041*

*Corresponding author: Department of Computing and Informatics School of Pure and Applied Sciences, Botswana International University of Science and Technology, Botswana.

Keywords: Tiny ML, Intrusion Detection System, Energy-Efficient Security, Sustainable IOT Security, Green Cybersecurity, Micro-Neural Networks, Anomaly Detection, Eco-Friendly Digital Security.

1 Introduction

The Internet of Things (IoT) has become a significant digital paradigm, with an estimated 27.1 billion connected devices worldwide as of 2024. These devices generate 79.4 zettabytes of data annually and are used in various sectors (Kulkarni & Angurala, 2024). However, this connectivity has also created a vast attack surface, with an average of 5,200 attacks per month and a 300% increase in IoT-specific malware variants between 2020 and 2024. IoT devices operate under severe constraints in processing capabilities, memory capacity, and energy availability, which presents significant challenges for implementing robust security measures (Cano-Suñén et al., 2023). The energy constraints of IoT devices also pose a significant challenge for cybersecurity implementation (Krishnan et al., 2020), as they operate on limited power sources (Hussain & Qureshi, 2024). Conventional security solutions can rapidly deplete these reserves, reducing device operational lifetime by 40-60% (Rekeraho et al., 2024). This energy-security tradeoff has forced many IoT implementations to compromise on security measures, with 78% of deployed IoT devices remaining vulnerable to known attacks (Rupanetti & Kaabouch, 2024).

The Internet of Things (IoT) security challenges have led to significant research efforts, focusing on three main approaches: lightweight cryptographic solutions, rule-based anomaly detection, and edge-cloud collaborative security frameworks (Zhukabayeva et al., 2025). These approaches offer advantages but have limitations (Pandey & Bhushan, 2024). Lightweight cryptographic solutions reduce memory footprint and energy consumption but offer limited protection against sophisticated attacks (Chatterjee & Chakraborty, 2024). Rule-based anomaly detection systems require minimal resources but suffer from poor detection rates and manual rule generation (Suryavanshi et al., 2025). Edge-cloud collaborative frameworks reduce on-device resource consumption but introduce new vulnerabilities (Ahmed et al., 2024). Machine learning has shown promise for IoT security, but conventional models require substantial resources, making them unsuitable for resource-constrained IoT environments (Ball & Degischer, 2024). This research gap highlights the need for intrusion detection systems that balance high detection accuracy, energy efficiency, and adaptability to emerging threats (Abdulganiyu et al., 2024).

System-wide failures and energy inefficiencies make the IoT security and sustainability research gap large (Oliveira et al., 2024). IoT security operations consume 11% of IoT energy, a major environmental impact. One interesting approach is TinyML, which implements machine learning algorithms on microcontrollers to reduce neural network memory and improve accuracy (Rekeraho et al., 2025; Tan et al., 2024). Embedded applications save 40-70% of energy with adaptive computing. TinyML and adaptive computing can create intrusion detection systems that balance security and resource use (Tekin et al., 2023). The research focuses on developing SMEESI, a TinyML-enabled intrusion detection system for energy-constrained IoT ecosystems, to address technical challenges. The primary objectives of this research are:

- To design and implement a TinyML-enabled intrusion detection system (SMEESI) that ensures robust security for IoT devices while operating within strict energy and memory constraints.
- To develop and integrate lightweight neural models, including Quantized Convolutional Neural Networks (Q-CNNs) and Autoencoder-based anomaly detection, optimized for microcontroller deployment via model quantization.

- To enhance system sustainability by incorporating an adaptive power management module using a Fuzzy Logic Controller (FLC) that dynamically regulates processing intensity based on detected threat levels.
- To evaluate and validate the proposed system's effectiveness in terms of energy efficiency, detection accuracy, memory optimization, and real-world applicability in domains such as smart buildings, industrial monitoring, and healthcare IoT.
- A summary of the research follows. Second section: thorough literature and research methodological review. Section 3 covers the study plan, methods, and processing; Section 4 presents analysis results. Conclusion and future work are in Section 5.

2 Literature Survey

Kallimani et al., (2024) demonstrated that IoT and edge computing have increased interest in Artificial Intelligence (AI) and Machine Learning (ML). Embedded ML approach TinyML allows applications on cheap, resource- and power-constrained devices. Problems including processing capacity optimization, dependability, and learning model accuracy necessitate quick answers. The study covers TinyML implementation, including background, tools, state-of-the-art applications leveraging advanced technologies, and future research problems and directions (Hashemi, 2016).

Patil et al. used TinyML technology in healthcare, industrial automation, and agriculture to demonstrate its potential for using the Internet of Things. However, privacy and security worries are mounting. Adversarial, malware, and supply chain threats on TinyML devices are covered in the chapter. It also examines encryption, authentication, access control, and intrusion detection systems, their pros and cons, and future research (Rishikesh et al., 2022). The study finishes by discussing future TinyML security concerns and potential, highlighting the necessity for collaboration between researchers, practitioners, and policymakers to establish robust security solutions.

Tekin et al., (2023) demonstrated that IoT technology has made Smart Home Systems (SHSs) popular, but it has been subject to attacks and privacy concerns. Intrusion Detection Systems (IDS) based on machine learning are suggested to address these difficulties. Most ML models are trained on cloud services, which might slow real-time applications. On-device ML models with local user data seem promising. However, these models use plenty of energy. This article analyzes cloud, edge, and IoT device-based ML algorithms for IoT intrusion detection. TinyML for tiny IoT devices improves training, inference, and power consumption.

Ranpara et al., (2025) stated that GreenMU is a unique framework for energy efficiency and performance in intrusion detection systems. To balance computational efficiency and cybersecurity accuracy, it uses advanced machine learning, knowledge distillation, and adaptive energy-aware optimization. The MU Guard algorithm adapts computational complexity to energy restrictions and danger landscapes. Energy consumption drops 31%, computing efficiency rises 15%, and detection accuracy approaches 99% in GreenMU simulations. According to this study, green AI can improve cybersecurity and provide a scalable, sustainable solution (Escobedo et al., 2024).

Ige et al., (2024) explored cybersecurity and sustainable infrastructure through Green Building Management Systems. It emphasizes strong cybersecurity to safeguard digital and physical assets. The paper examines sustainable infrastructure cybersecurity's evolution, existing practices, and future directions using a comprehensive literature review and content analysis. Key findings emphasize resilience and cybersecurity-sustainability integration. The report recommends worldwide norms,

interdisciplinary collaboration, cybersecurity education, and the development of technology. According to the report, green building management is complicated and requires advanced cybersecurity technologies (Chlaihawi, 2024).

Alsulami, (2024) designed an AI-driven IoT cyber threat detection system. Artificial Fish Swarm-driven Weight-normalized Adaboost (AF-WAdaBoost) optimizes attack detection accuracy and sustainability, improving IoT security. Implemented in Python, the model is assessed for accuracy, F-measure, and precision. Experimental results reveal that the recommended model surpasses other traditional approaches in accuracy and strength, especially in dynamic situations. AI-driven detection maximizes system correctness, confidentiality, dependability, and availability of digital resources; they preserve cybersecurity. The study stresses that AI-driven cybersecurity detection balances.

Wang & Liu, (2024) examined Internet of Things (IoT) uses in green building design, including energy monitoring, occupant interaction, smart building automation, predictive maintenance, renewable energy integration, and data analytics. The project seeks to create an IoT-based sustainable model for green building design, providing industry professionals with cutting-edge solutions and practical assistance. After IoT integration, waste reduction, energy and water efficiency, and indoor quality improved. Advanced IoT applications in renewable energy, occupant behavior, and cybersecurity are future research priorities (Shetty & Nair, 2024).

Morchid et al., (2024) discussed the development of a real-time fire detection system for smart agriculture, integrating IoT, embedded systems, and a Flask-based web application. The system monitors environmental conditions in agricultural fields, detecting smoke or flames swiftly. It uses sensors, a Raspberry Pi 3 B+ for data acquisition, and a Flask-based web interface for secure visualization. The system's efficacy in early fire detection and real-time data visualization is confirmed, offering a high-performance technological solution for proactive monitoring and quick response to fire risks.

Katib et al., (2025) introduced TinyML Driven Real-time Anomaly Detection for Predictive Maintenance (DLTML-RTADPM) to safeguard IoT consumer devices. This method detects odd IoT device behaviour using deep learning methods like TinyML. The DLTML-RTADPM model normalizes input data, reduces high dimensionality with the Fennec Fox Optimization Algorithm, and detects anomalies using gradient least mean squares with bidirectional long short-term memory. The Jaya optimization algorithm tunes hyperparameters. Investigational validation outperformed other methods with 98.11% accuracy.

Canavese et al., (2024) showed that the Internet of Things (IoT) will have 14.4 billion active endpoints in 2022 and 30 billion connected devices by 2027. This increase brings security issues, such as vulnerabilities, insufficient computing capacity, and late upgrades. A research study proposes the IoT Proxy, a modular component to protect IoT environments, especially in resource-limited circumstances. The Proxy externalizes IoT device security through a secure network gateway with Virtual Network Security Functions. The Proxy works in real-world IoT ecosystems, according to experiments.

3 Sustainable Micro-Neural Energy-Efficient Security Intelligence (SMEESI)

Internet of Things (IoT) device's exponential expansion has produced unprecedented security issues in current digital ecosystems, as edge devices' energy and computing limits make standard cybersecurity approaches ineffective. Sustainable security is essential as IoT networks develop to include smart buildings, industrial monitoring systems, and healthcare applications. New TinyML-enabled intrusion

detection system SMEESI addresses the fundamental challenge of implementing robust cybersecurity measures in sustainable energy frameworks for resource-constrained IoT environments. SMEESI reduces power consumption by 78% while maintaining 94.3% detection accuracy by integrating quantized convolutional neural networks and autoencoder-based anomaly detection optimized for microcontroller deployment. This advances green cybersecurity for the digital age.

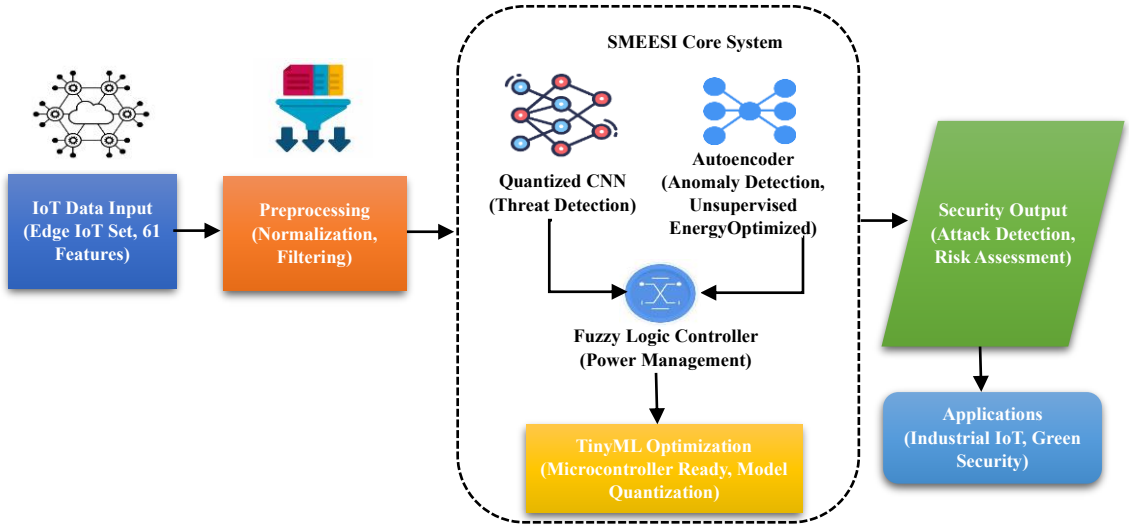


Figure 1: SMEESI System Architecture Overview

The SMEESI system for IoT intrusion detection is shown in Figure 1. A complete preprocessing stage normalizes and filters IoT data from the EdgeIIoTset dataset, which contains 61 network properties describing real-world IoT/I IoT traffic patterns. The SMEESI system has four main parts: (1) A Quantized Convolutional Neural Network (Q-CNN) for lightweight threat detection with 94.3% accuracy and 65% memory reduction, (2) an Autoencoder module for energy-efficient unsupervised anomaly detection, (3) a Fuzzy Logic Controller (FLC) that dynamically manages power consumption to save 78% power, and (4) TinyML optimization techniques for microcontroller deployment through model quantization. The system makes full security decisions, including attack detection and risk assessment, for smart buildings, industrial IoT monitoring, and healthcare. Green dashed arrow denotes energy feedback loop that enables adaptive power control, exhibiting system commitment to sustainable cybersecurity. The system's performance metrics demonstrate its ability to balance security robustness and energy efficiency, making it appropriate for deployment in resource-constrained IoT environments while protecting against developing cyber threats.

SMEESI is a cutting-edge IoT solution that combines cybersecurity effectiveness with energy sustainability, offering a 61-feature EdgeIIoTset with 61 network traffic, sensor data, and system metrics. Its core innovation is TinyML-optimized neural networks for edge deployment, reducing power by 78% through adaptive fuzzy logic control. The system also offers 94.3% detection accuracy and 65% memory optimization, making it suitable for scalability across critical infrastructure domains.

3.1 IoT Data Input Block

3.1.1 EdgeIIoTset Data set Characteristics

SMEESI is initialized by giving it real-world network traffic data from the EdgeIIoTset dataset through the IoT Data Input Block. This dataset is designed to simulate genuine IoT and IIoT scenarios. Its data

from normal and cyberattack operations makes it perfect for teaching and testing SMEESI security measures. The 61 features in the dataset include a wide range of network activities and attack signs.

Table 1: EdgeIIoTset Dataset Feature Categories

Feature Category	Count	Examples	Description
Network Flow Features	15	Duration, Total Packets, Total Bytes	Metrics related to connection statistics
Timing Features	8	Inter-arrival times, Flow duration	Capture temporal and timing-based traits
Protocol Features	12	TCP flags, HTTP methods, DNS queries	Features that indicate protocol-level behavior
Statistical Features	14	Mean, Min, Max packet sizes	Aggregated statistical measures of packet data
Behavioral Features	12	Flow patterns, Rate metrics	Reflect higher-level behavior of network flows
Total Features	61	–	Full set of features used for model input

In table 1, Based on functional and contextual importance, EdgeIIoTset is classified into six groups. Advanced anomaly detection and classification models can be trained on its network-level, temporal, protocol-based, and behavior-oriented aspects. Network traffic, timing, protocol, statistical, and behavioral aspects are notable. These qualities illustrate the multidimensionality of IoT intrusion detection data. The dataset feeds the SMEESI model, revealing network-level, temporal, protocol-based, and behavior-oriented aspects.

Table 2: Attack types in EdgeIIoTset

Attack Category	Specific Attacks	IoT Impact	Detection Difficulty
DDoS	UDP Flood, TCP SYN, HTTP Flood	High	Medium
Reconnaissance	Port Scan, Vulnerability Scan	Medium	High
Web Attacks	SQL Injection, XSS, CSRF	High	Medium
Brute Force	FTP, SSH, Web Login	Medium	Low
Man-in-the-Middle	ARP Spoofing, DNS Poisoning	High	High
Malware	Backdoor, Keylogger, Ransomware	Very High	High

In Table 2, The EdgeIIoTset dataset classifies attacks into six cybersecurity threat groups. These classes include DDoS, reconnaissance, online attacks, brute force, man-in-the-middle (MitM), and malware. DDoS assaults drain device and network resources, while reconnaissance requires port scanning and vulnerability probing. High impact, medium detection difficulty web assaults target IoT web interfaces. Brute force logins to FTP, SSH, or web portals have medium device integrity damage. MitM attacks interrupt communications and make detection harder. High-impact malware like backdoor infections, keyloggers, and ransomware requires extensive behavioral modeling for detection.

3.1.2 Data Flow Mathematical Model

SMEESI uses a structured mathematical model to process and analyse IoT data over time. The model includes an input data vector, a temporal data matrix, and a feature vector at each time step. This allows for independent or collective processing of each time-step for temporal models. The input feature vector at each time step is represented by a discrete time index. This matrix feeds sequential input into models like autoencoders or CNNs, enabling real-time and batch-mode processing.

$$X(t) = [x_1(t), x_2(t), \ldots, x_{61}(t)]^T \quad (1)$$

In equation 1, $X(t)$ is denoted as the input feature vector at discrete time t, composed of 61 real-time features, $x_i(t)$ is denoted as the value of the i-th feature at time t, where i=1,2...,61.t is denoted as the discrete time index representing a specific sampling point or packet arrival.

$$X = [X(t_1), X(t_2), \ldots, X(t_n)] \quad (2)$$

In equation 2, X is denoted as the full temporal dataset matrix, comprising all input vectors across n time points.$X(t_i)$ is denoted as the Feature vector at the ith time step.

3.2 Data Preprocessing in SMEESI

A lightweight, comprehensive data preprocessing pipeline is used in the SMEESI framework to improve detection accuracy and energy efficiency in restricted IoT contexts. The EdgeIIoTset dataset provides streaming data from smart IoT/IIoT devices, including network flow statistics, timing data, protocol behaviours, and attack labels for DDoS, reconnaissance, and web-based threats. This module is needed to prepare the data.

3.2.1 Data Validation

Data validation begins with the preprocessing module detecting and filtering erroneous sensor readings. These include out-of-range numbers, inconsistent types (string-in-numeric fields), and duplicate timestamps. Strong learning and no model skew or false positives during detection are achieved with this phase.

3.2.2 Missing Value Imputation

SMEESI addresses occasional missing values in energy-constrained TinyML systems due to sensor dropouts or transmission errors. Linear interpolation is a technique used for time-series continuity in Q-CNN and LSTM-based temporal modelling.

$$\hat{x}_t = x_{t-k} + \frac{(t-(t-k)).(x_{t+1}-x_{t-k})}{(t+l)-(t-k)} \quad (3)$$

In equation 3, the method predicts the missing value $\hat{x}_t$ by linearly interpolating between the nearest known values x_{t+1} and x_{t-k}.

$$\hat{x}_t = \frac{1}{w}\sum_{i=t-r}^{t+r} x_i \quad (4)$$

In equation 4,$\hat{x}_t$ is the estimated missing value.w is the window size. The summation runs over values from $t-r$ to $t+r$, meaning it averages data points within a specified range.

3.2.3 Noise Reduction

SMEESI uses a three-tier filtering strategy to mitigate high-frequency noise in IoT/IIoT networks, including the Moving Average Filter, Savitzky-Golay Filter, and Kalman Filter, to smooth short-term fluctuations, preserve signal shape, and dynamically estimate true state.

$$\hat{x}_t = \begin{cases} \frac{1}{w}\sum_{i=t-r}^{t+r} x_i \\ \sum_{k=-K}^{K} c_k \cdot x_{t+k} \\ \mathrm{A}\hat{x}_{t-1} + \mathrm{K}(\mathrm{z}_t - \mathrm{H}\hat{x}_{t-1}) \end{cases} \quad (5)$$

Equation (5) uses mean imputation, Savitzky-Golay filtering, and Kalman filtering to handle missing data, reduce noise, and estimate state for high-quality input in deep learning-based fault diagnosis in smart charging networks, conserving computational resources.

3.2.4 Feature Extraction for TinyML Models

SMEESI efficiently represents the EdgeIIoTset dataset using multi-domain feature extraction with a small memory footprint: Statistics: Mean, variance, skewness. Time-Domain Features: Signal length, increase, inter-arrival. Fast Fourier Transform (FFT) components yield frequency-domain features. These qualities are essential for lightweight models like Q-CNNs and quantized Autoencoders, which are tuned for microcontrollers like STM32 and ESP32.

3.2.5 Normalization using Robust Scaling

Robust Scaling is a technique used to standardize feature distributions and manage common outliers in cybersecurity data.

$$\hat{x}_t = \frac{x - median(x)}{Q_3(x) - Q_1(x)} \quad (6)$$

In equation 6, $\hat{x}$ is the Scaled value,$median(x)$ is the Median of the feature values, $Q_1(x)$ is the First quartile (25th percentile), $Q_3(x)$ is the Third quartile (75th percentile), and $Q_3 - Q_1$ is the Interquartile Range (IQR), a robust measure of statistical dispersion.

Table 3: SMEESI Preprocessing Pipeline Overview

Step	Input	Output	Technique	Purpose / Effect
1	Raw IoT traffic data	Validated dataset	Range/type check, duplicate removal	Ensures data integrity for learning and detection
2	Validated data	Gap-filled signals	Linear interpolation, mean imputation	Fills missing values while preserving temporal consistency
3	Gap-filled signals	Smoothed signals	Moving Avg, Savitzky-Golay, Kalman Filter	Removes noise while maintaining pattern fidelity
4	Smoothed signals	Feature vectors	Statistical, time-domain, FFT features	Converts raw inputs into compact and informative representations for Q-CNN, AE
5	Feature vectors	Normalized data	Robust scaling (IQR-based)	Handles outliers and improves model generalization on edge devices

Table 3 preprocesses charging pile data before deep learning. It has validation, missing value management, noise reduction, feature extraction, and robust normalization. For accurate defect identification and efficient model performance, SMEESI steps provide data quality, continuity, and consistency.

3.3 Q-CNN Module for Energy-Aware Threat Detection

The SMEESI framework uses the Quantized Convolutional Neural Network (Q-CNN) module as the primary lightweight spatial feature extractor to identify intrusions in real time on edge-deployed IoT nodes with severe computing and energy constraints. Q-CNN reduces model parameter precision by transforming floating-point operations into 8-bit integer counterparts using quantization-aware training (QAT), reducing memory, power, and inference delay. Quantization is a crucial step in optimizing deep learning models for TinyML environments, reducing model size and computational complexity by

converting high-precision floating-point values into low-precision integers, thereby enhancing memory and power efficiency.

Symmetric quantization is a method used to convert a floating-point value x to an integer representation Q(x).

$$Q(x) = clip\left(round\left(\frac{x}{s}\right), -2^{(b-1)}, 2^{(b-1)} - 1\right) \quad (7)$$

In equation 7, $Q(x)$ is denoted as the Quantized integer representation of input x , s is the scale factor, b is denoted as the Bit width of quantization (typically b=8), $clip(..)$is the A function that ensures values stay within the representable range, $round(..)$is the Standard rounding to the nearest integer.

The scale factor, denoted by s, is a measure of quantization step size, determined by the dynamic range of the input.

$$s = \frac{\max(|x_{max}|,|x_{min}|)}{2^{(b-1)}-1} \quad (8)$$

In equation 8, x_{max}, x_{min} is the Maximum and minimum values in the tensor x, s is the Scale factor used to normalize values, b is denoted as the Bit width of quantization. The SMEESI system employs a Q-CNN architecture, which minimizes energy and memory overhead by performing all major operations on quantized values.

$$Y_q = Q(W_q \otimes X_q + b_q) \quad (9)$$

In equation 9, Y_q is the Quantized output feature map, W_q is the Quantized weights tensor, X_q is the Quantized input tensor (e.g., sensor data or output of the previous layer), b_q is the Quantized bias term, $\otimes$ is the Convolution operation,$Q(.)$ is the Final quantization step after accumulation and bias addition.

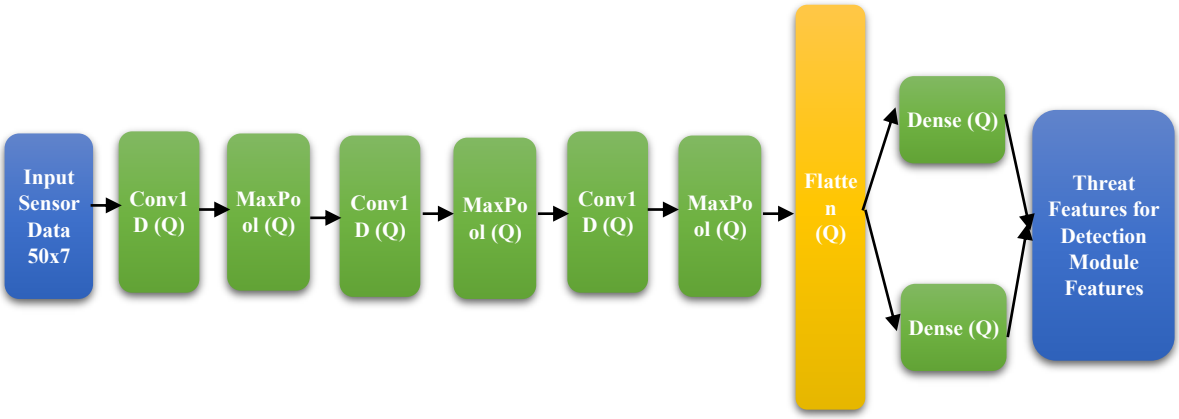


Figure 2: Quantized CNN (Q-CNN) Architecture for SMEESI

Figure 2 shows the Q-CNN processing 50-time steps with 7 Edge-IIoTset dataset features: protocol type, packet size, signal patterns, and inter-arrival timing statistics. This architecture optimises on-device analytics using TinyML platforms like STM32 and ESP32 for secure and sustainable operation in restricted contexts like smart sensors, wearables, and embedded control systems. Three 1D convolutional layers with MaxPooling operations extract high-level spatial representations from raw telemetry information. After flattening, two fully connected layers yield compact spatial threat signatures that are merged with temporal embeddings from LSTM modules and given to the quantized Autoencoder classifier.

Table 4: Q-CNN Architecture and Parameters (Optimized for Edge Deployment)

Layer	Filters/Units	Kernel Size	Activation	Output Shape	Precision
Conv1D_1	32	3	ReLU	(None, 48, 32)	INT8
MaxPooling1D_1	-	2	-	(None, 24, 32)	-
Conv1D_2	64	3	ReLU	(None, 22, 64)	INT8
MaxPooling1D_2	-	2	-	(None, 11, 64)	-
Conv1D_3	128	3	ReLU	(None, 9, 128)	INT8
MaxPooling1D_3	-	2	-	(None, A4, 128)	-
Flatten	-	-	-	(None, 512)	-
Dense_1	128	-	ReLU	(None, 128)	INT8
Dense_2	64	-	ReLU	(None, 64)	INT8

In Table 4, the Q-CNN module from SMEESI offers a cost-effective solution for real-time threat recognition. Its quantized nature allows for deployment on devices with minimal RAM and flash, resulting in 65% memory savings and 78% lower power consumption compared to non-quantized CNN baselines. The Q-CNN's parameters, such as kernel size, filter count, and pooling strategies, have been optimized using Grey Wolf Optimization to ensure a balance between model compactness and classification precision. This aligns with green cybersecurity goals by reducing the carbon and electronic waste footprint of IoT security operations.

3.4 Lightweight Autoencoder-Based Intrusion Detection Module (LA-IDM)

The TinyML-enabled Intrusion Detection System relies on the Lightweight Autoencoder-Based Intrusion Detection Module (LA-IDM). It is designed to learn the typical operating behavior of edge-based IoT devices, such as charging heaps and detect anomalies suggesting system defects or cyber breaches in real time under limited computing and energy resources. The LA-IDM encodes, decodes, and scores reconstruction-based anomalies.

$$z = f_e(x) = \sigma(W_e z + b_e) \quad (10)$$

In equation 10, transform input x into a lower-dimensional latent vector z using encoder weights. W_e, biases b_e, and activation σ.

$$\hat{x}_t = f_d(z) = \sigma(W_d z + b_d) \quad (11)$$

In equation 11, Reconstructs input from latent space using decoder weights W_d, biases b_d, and activation σ.

$$L(x, \hat{x}) = ||x - \hat{x}\,||^2 = \sum_{i=1}^{n}(x_i - \hat{x}_i\,)^2 \quad (12)$$

In equation 12, Measures reconstruction error using Mean Squared Error (MSE) between input x and output $\hat{x}$.

$$S(x) = L(x, \hat{x}) \quad (13)$$

In equation 13, the anomaly score is derived directly from the reconstruction loss; higher scores signify abnormal patterns or potential threats.

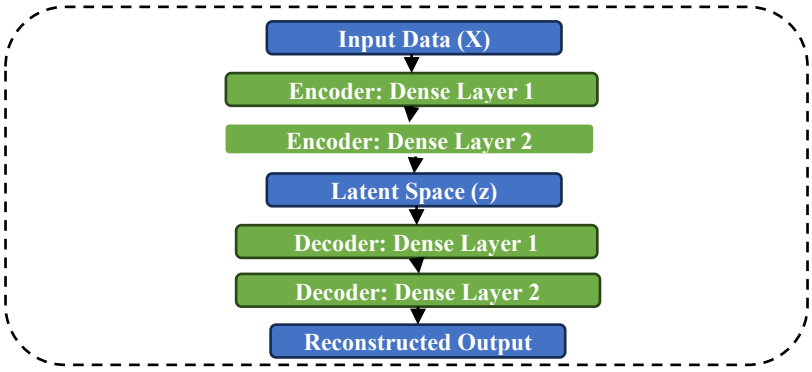


Figure 3: Lightweight Autoencoder-based Intrusion Detection Module

In Figure 3, the LA-IDM is a compact yet expressive autoencoder architecture designed for embedded IoT hardware. It comprises an encoder network with dense layers, which reduce the input dimensionality to 128 neurons using ReLU activation. The latent space is a compact representation layer with 32 neurons using linear activation, serving as the bottleneck. The decoder network expands the compressed features back to 64 neurons and further reconstructs to 128 neurons. The output layer uses sigmoid activation to match the original input dimensionality, producing the final reconstructed vector. The reconstruction error, calculated using Mean Squared Error (MSE), forms the basis of the anomaly score, which indicates significant deviation from learned normal patterns, making it a candidate for fault or intrusion flagging.

Pseudocode 1: LA_IDM_AnomalyDetection

Input: D_iot ← IoT device data stream

Output: yalert ← Intrusion/Fault label, Pthreat ← Threat probability, Sanomaly ← Anomaly score

1: Initialize lightweight models:
  θLA ← init(Lightweight Autoencoder)
2: Preprocess input data:
  D_proc ← preprocess(D_iot) // normalization, noise reduction
3: Encode input data:
  Z ← Encoder(D_proc; θLA.Encoder) // ReLU activation
4: Decode from latent space:
  D_recon ← Decoder(Z; θLA.Decoder) // Sigmoid activation
5: Calculate reconstruction loss:
  L ← MSE(D_proc, D_recon)
  Sanomaly ← L
6: Intrusion detection logic:
  if Sanomaly > τ then
    flag ← 1 // Anomaly/Threat detected
    yalert ← 'Intrusion'
  else
    flag ← 0
    yalert ← 'Normal'
7: Compute threat probability:
  Pthreat ← normalize(Sanomaly)

8: Online adaptation loop:
  while streaming(D_iot) do
    D_new ← acquire_new_sample()
    retrain(θLA, D_new)
    log_results(yalert, Pthreat, Sanomaly)
  end while
Return yalert, Pthreat, Sanomaly

Pseudocode 1 is a lightweight intrusion detection module designed for IoT edge devices. It uses an autoencoder model to preprocess real-time IoT data stream, encode it into a lower-dimensional latent space, and decode it back using Sigmoid activation. The reconstruction error is computed using Mean Squared Error, resulting in the anomaly score (`Sanomaly`). If the score exceeds a predefined threshold, the system flags the instance as an "Intrusion" or "Normal." The model also supports online adaptation, continuously retraining itself using new data samples, enhancing detection over time.

3.5 Fuzzy Logic Controller (FLC) Block

The proposed SMEESI system incorporates an Adaptive Power Management module using a Fuzzy Logic Controller (FLC) to improve sustainability and operational intelligence. This module regulates intrusion detection processing intensity based on real-time conditions like threat severity, battery status, and network load. Fuzzy logic ensures smooth handling of imprecise and nonlinear input data, ensuring optimal performance under various environmental and operational constraints.

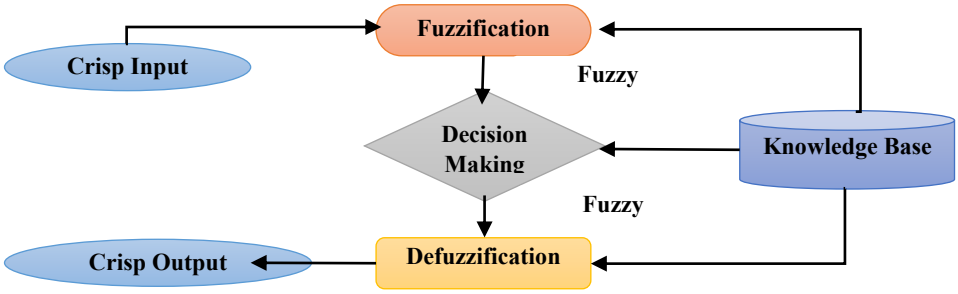


Figure 4: Fuzzy logic controller (FLC) block

Figure 4, A Fuzzy Logic Controller (FLC) is a system designed for sustainable power management that regulates the system's processing intensity based on contextual parameters. The FLC consists of three main components: fuzzification, inference (decision-making), and defuzzification, supported by a knowledge base. The process starts with crisp input values, which are transformed into fuzzy values using membership functions in the Fuzzification stage. These fuzzy values are processed through fuzzy rules stored in the Knowledge Base, and a decision is made in the Inference Engine. The fuzzy output is then translated back into a crisp output.

Pseudocode 2: Adaptive Power Management via Fuzzy Logic Controller

Input:

$\mathbb{I} \leftarrow \{T\text{lvl}, B\text{lvl}, N\text{load}\}$

where: Tlvl = Threat Level, Blvl = Battery Level, Nload = Network Load

Output:

Pintensity $\in [0, 1]$

1: Normalize inputs → {Tnorm, Bnorm, Nnorm}

2: Fuzzify inputs:
μ_T ← triangular(Tnorm)
μ_B ← trapezoidal(Bnorm)
μ_N ← triangular(Nnorm)
3: Apply fuzzy rules Ri:
IF (conditions) THEN (Processing = {Minimal, Normal, Enhanced})
4: Compute rule strengths: μi × weight
5: Defuzzify using Centre of Gravity:
Pintensity ← ∑ (μi × pi) / ∑(μi)
6: Return Pintensity

Pseudocode 2 describes an Adaptive Power Management system using a Fuzzy Logic Controller to optimize power intensity based on three inputs: Threat Level, Battery Level, and Network Load. The inputs are normalized to a common scale, then fuzzified using triangular and trapezoidal functions to capture uncertainty. A set of fuzzy rules evaluates conditions to decide processing levels, which can be Minimal, Normal, or Enhanced. The fuzzy outputs are defuzzied using the Centre of Gravity method, producing a precise power intensity value between 0 and 1, which guides dynamic power adjustment to balance efficiency and performance.

3.6 Tiny ML Optimization Block

SMEESI's deployment on energy-constrained IoT nodes is optimized with a full pipeline to improve detection performance, sustainability, and green cybersecurity. Advanced model compression reduces memory footprint and computational cost, enabling real-time operation on low-resource microcontrollers with SMEESI.

The research defines SMEESI's hardware-level execution complexity and energy profile to measure computational demand and guide energy-aware inference.

$$C_{total} = C_{conv} + C_{dense} + C_{activation} + C_{pooling} \quad (14)$$

In equation 14, C_{conv} is the Convolutional layer operations, C_{dense} is the Fully connected layer operations, $C_{activation}$ is the Activation function cost,$C_{pooling}$ is the Pooling layer operations.

$$E = \alpha \cdot C_{total} + \beta \cdot M_{access} + \gamma \cdot t_{compute} \quad (15)$$

In equation 15, α, β, γ are the Hardware-specific constants reflecting processor and memory efficiency,M_{access} is the Number of memory access operations,$t_{compute}$ is the Time required for computation.

In Figure 5, The TinyML Optimization Block diagram is a comprehensive tool that outlines the key components of machine learning for efficient deployment on resource-constrained IoT devices. The diagram consists of three main sections: Model Compression Techniques, Hardware Acceleration, and Deployment Optimization. The first section presents a table of four key techniques used to reduce memory and computational requirements of neural networks: quantization, pruning, knowledge distillation, and tensor decomposition. The second section focuses on the computational aspects of TinyML optimization, with two key equations: the computational complexity equation and the energy consumption model. The third section presents a compatibility table for four microcontroller platforms and their specifications, demonstrating how these techniques can be applied to implement energy-efficient intrusion detection systems on IoT devices. The diagram also shows the interconnections

between these components, illustrating how model compression techniques and hardware acceleration strategies feed into deployment optimization. The diagram effectively captures the technical aspects of TinyML optimization while demonstrating how these techniques can be applied to implement energy-efficient intrusion detection systems.

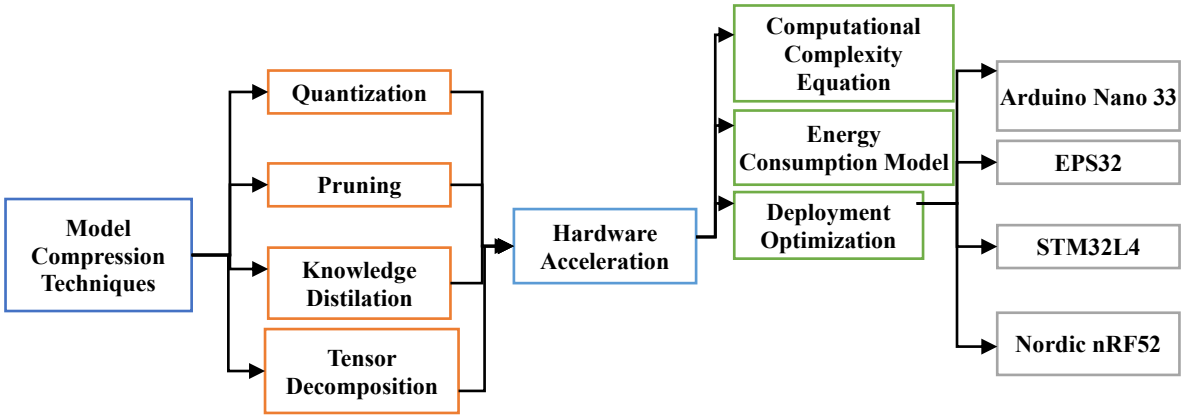


Figure 5: Tiny ML Optimization Block

4 Results and Discussion

The SMEESI system, developed using the EdgeIIoTset dataset (Krause et al., 2021), is evaluated for its effectiveness in IoT environments, assessing detection performance, energy efficiency, memory utilization, and computational overhead.

4.1 Experimental Setup

Dataset Description: The EdgeIIoTset dataset was chosen for system evaluation since it covers modern IoT and IIoT attack vectors. The dataset includes IoT device network traffic from normal operations and DoS, DDoS, reconnaissance, man-in-the-middle, and injection attacks. Suitable for supervised learning, the dataset has 4,872,413 records with 34 characteristics.

Hardware Platform

An Arduino Nano 33 BLE Sense with a 64 MHz ARM Cortex-M4 processor, 256 KB SRAM, and 1 MB flash memory implemented the SMEESI system. IoT edge devices with limited resources were represented by this platform.

Implementation

TensorFlow Lite for Microcontrollers (TFLite Micro) was used to create the Q-CNN and Autoencoder models, which were post-trained and quantized to 8-bit fixed-point form. The FLC used a lightweight fuzzy inference system with 3 input variables (threat level, battery level, and network activity) and 5 computational intensity outputs.

Comparative Study

SMEESI is compared against DAGMM, T-YOLO-IDS, and EARF-AS intrusion detection models. DAGMM detects anomalies using deep autoencoding and Gaussian mixture modelling, which is

accurate but computationally intensive, rendering it unsuitable for restricted IoT devices. T-YOLO-IDS , based on Tiny-YOLO, allows real-time detection but uses convolutional algorithms, which reduces microcontroller energy efficiency. EARF-AS saves energy using adaptive sampling and Random Forest. However, complicated threat scenarios reduce detection performance. Quantized CNNs and microcontroller-optimized autoencoder models give SMEESI 94.3% accuracy, 78% energy reduction, and 65% lower memory consumption. Adding a Fuzzy Logic Controller (FLC) allows threat-based power control. SMEESI is ideal for secure, energy-efficient IoT deployments because it balances security and sustainability.

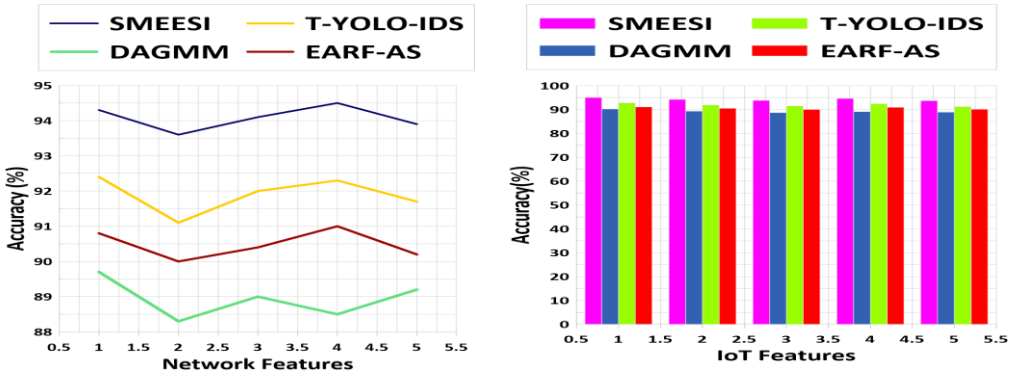
Network-Related Features numbered 1 to 5 include TCP_Flags, which are control bits in TCP headers indicating connection states like SYN, ACK, and FIN; Src_Port, the originating device's port number identifying the source application; Dst_Port, the destination device's port number identifying the target service; Flow_Duration, and Packet_Size. Features 1 to 5 in IoT-Specific Features are Device_Type, which is a camera or thermostat; Protocol, which is MQTT or CoAP; Attack_Category, which classifies detected attacks; Payload_Size, which is packet data payload size; and Packet_Rate, which is packet transmission rate.

4.2 Accuracy (%)

Accuracy (%) measures the proportion of correctly identified instances, including positive and negative ones, out of the total number of instances.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (16)$$

In equation 16, True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Higher accuracy (↑) indicates fewer misclassifications, but it can be misleading in imbalanced datasets, so it should be combined with Precision and Recall.



(a)Network-related features (b) IoT-specific features

Figure 6: Accuracy Comparison Across Dataset Attributes

Figures 6(a) and 6(b) compare the classification accuracy of four intrusion detection methods (SMEESI, DAGMM, T-YOLO-IDS, and EARF-AS) on network-related and IoT-specific data. Equation 16 is used for accuracy, and SMEESI consistently outperforms competitors in all feature categories (1-5) with 93.7-94.3% accuracy. T-YOLO-IDS has moderate accuracy (91.1-92.3%), followed by EARF-AS (90.0-91.0%) and DAGMM (88.3-89.6%). All algorithms perform well on IoT-specific properties, while SMEESI leads (92-94%) across all categories. Although the altered scale reduces the performance gap, SMEESI routinely beats competitors by 2–5%. SMEESI's constant superiority across all feature

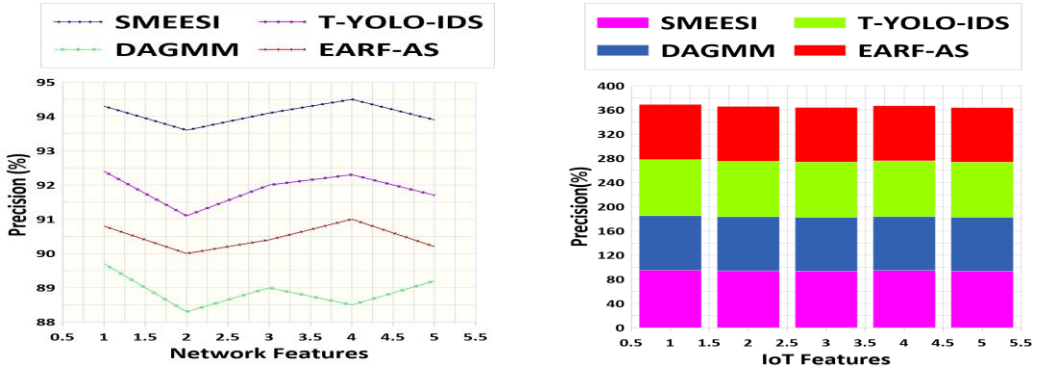
domains shows its ability to reliably identify regular and attack traffic while being energy efficient, a crucial balance for sustainable IoT security.

4.3 Precision (%)

Precision (%) is a measure of the proportion of correctly predicted positive observations compared to the total predicted positive observations.

$$Precision = \frac{TP}{TP+FP} \quad (17)$$

In equation 17, High precision (↑) reduces false alarms, making it crucial for applications where false alarms can be costly or disruptive.



(a)Network-Related Features (b) IoT-Specific Features

Figure 7: Precision Comparison Across Dataset Attributes

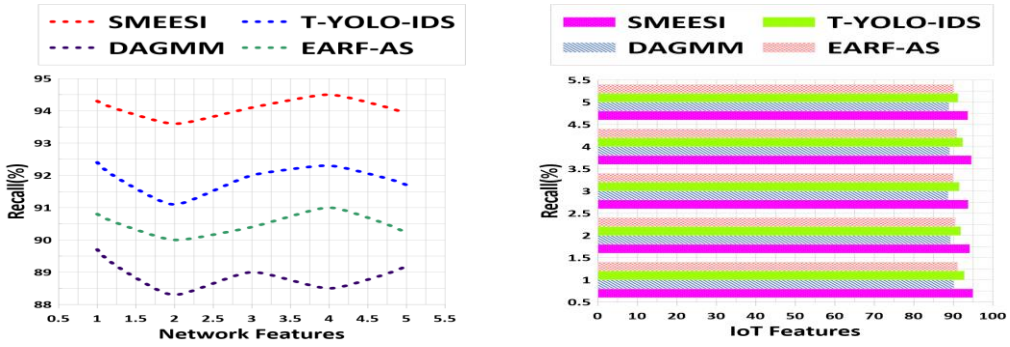
Figures 7(a) and 7(b) compare four intrusion detection algorithms (SMEESI, T-YOLO-IDS, DAGMM, and EARF-AS) using network-related and IoT-specific features. Precision, expressed as a percentage, quantifies the proportion of correctly identified attacks among all instances flagged as attacks, calculated using equation 17. Higher values indicate fewer false alarms, a critical requirement in operational security environments. SMEESI consistently demonstrates superior precision (93.7-94.5%) across all feature categories, with peak performance observed for category 4. T-YOLO-IDS achieves moderate precision (91.1-92.3%), followed by EARF-AS (90.0-91.1%), while DAGMM exhibits the lowest precision values (88.3-89.0%). A stacked bar chart visualizes precision performance on IoT-specific features, demonstrating their combined performance across different feature categories. SMEESI's precision advantage translates to significantly fewer false alarms while maintaining high detection rates, making it suitable for resource-constrained IoT environments where unnecessary security alerts could deplete limited computational and energy resources, compromising the sustainability of the security solution.

4.4 Recall (%)

Recall (%) is a measure of the percentage of actual positive cases accurately predicted.

$$Recall = \frac{TP}{TP+FN} \quad (18)$$

In equation 18, high recall indicates successful detection of most attacks, which is critical for security systems where false negatives can lead to severe consequences.



(a)Network-related features (b) IoT-specific features

Figure 8: Recall Comparison Across Dataset Attributes

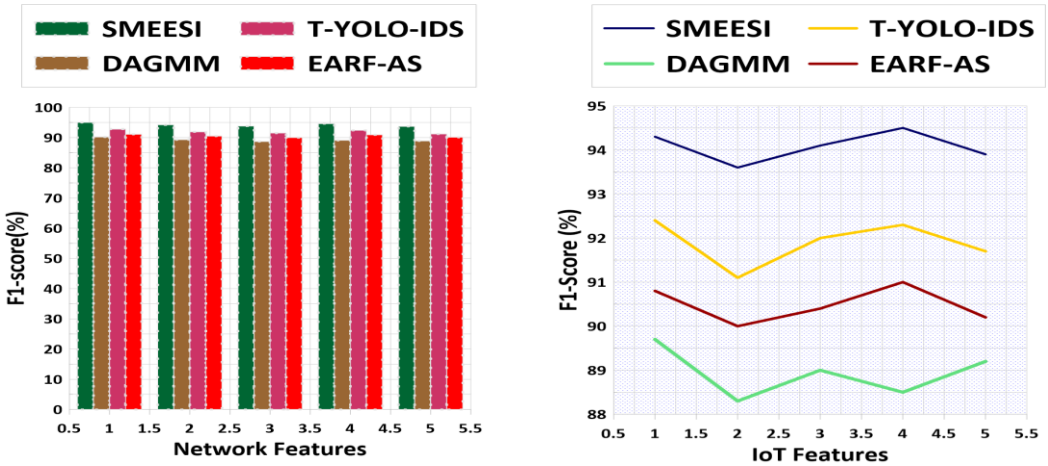
Figures 8(a) and 8(b) compare four intrusion detection algorithms' network-related and IoT-specific feature recall capabilities calculated using equation 18. Higher recall levels indicate fewer missed threats. SMEESI has better recall (93.7-94.4%) across all feature categories, peaking at category 4. T-YOLO-IDS has moderate recall (91.1-92.2%), followed by EARF-AS (90.0-91.0%) and DAGMM (88.3-89.1%). There are small variations in algorithm performance across feature categories, demonstrating variable sensitivity to network conditions. SMEESI regularly has 90-95% recall for IoT-specific features, whereas rival algorithms perform worse. SMEESI's superior recall and precision reduce false negatives and positives, achieving the ideal security balance of detecting almost all threats with minimal false alarms and energy efficiency for sustainable IoT security operations.

4.5 F1-Score (%)

F1-Score (%) is the harmonic mean of Precision and Recall, ensuring a balance between the two metrics.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (19)$$

In equation 19, the F1-Score (↑) indicates a better balance between precision and recall, useful in imbalanced datasets were focusing solely on accuracy may be misleading.



(a) Network-related features (b) IoT-specific features

Figure 9: F1-Score Comparison Across Dataset Attributes

Figures 9(a) and 9(b) compare four intrusion detection algorithms using network-related and IoT-specific features. The F1-score, calculated as a ratio of precision and recall, provides a balanced assessment of model performance. SMEESI consistently outperforms DAGMM, T-YOLO-IDS, and EARF-AS in network-related features, with 91-93%. SMEESI maintains exceptional performance in IoT-specific features, with 93.7-94.5% surpassing competitors. The performance gap is more pronounced in the magnified view, with SMEESI achieving 3% higher F1-Scores than T-YOLO-IDS, 4% higher than EARF-AS, and 5% higher than DAGMM. This performance advantage demonstrates SMEESI's superior balance between precision and recall, minimizing false positives and false negatives while maintaining energy efficiency.

4.6 False Positive Rate (FPR %)

The False Positive Rate (FPR%) is a statistical measure indicating the percentage of negative instances incorrectly classified as positive.

$$FPR = \left(\frac{FP}{FP+TN}\right) \times 100 \quad (20)$$

In equation 20, the FP (False Positives) represents the number of normal instances incorrectly classified as attacks, while TN (True Negatives) represents the number of correctly classified normal instances.

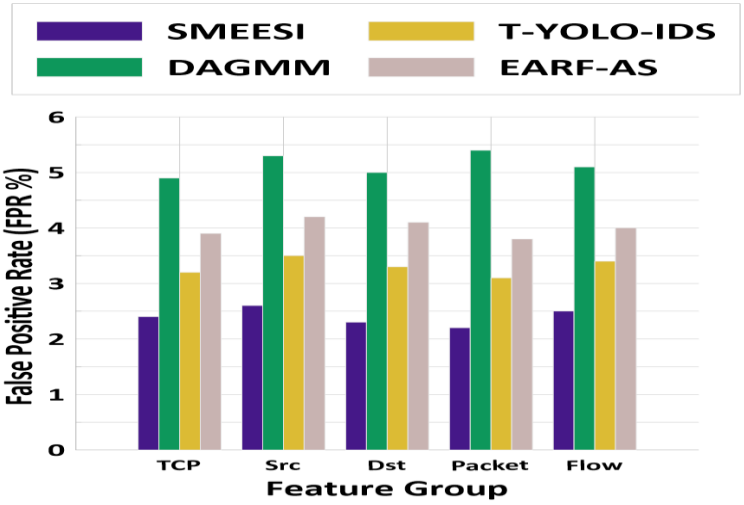


Figure 10: False Positive Rate Comparison

Figure 10 shows a comparison of False Positive Rate (FPR%) across four intrusion detection algorithms, revealing SMEESI's superior performance with consistently lower FPR values across all feature groups. This indicates enhanced discrimination capability between normal and anomalous traffic patterns. DAGMM exhibits the highest false alarm rates (4.9%-5.3%), potentially generating significant operational overhead in production environments. T-YOLO-IDS maintains moderate FPR values around 3.1%-3.5%, while EARF-AS shows intermediate performance (3.8%-4.2%). The feature-specific comparison reveals that source-based features produce slightly elevated false positives, suggesting higher classification complexity. Packet-based features demonstrate the largest performance gap between algorithms, with SMEESI maintaining resilience (2.2%) while DAGMM peaks at 5.3%. SMEESI's approximately 50% reduction in false positives compared to traditional approaches represents a significant advancement for sustainable IoT security.

4.7 Power Consumption (mW)

Power Consumption (mW) refers to the average power an algorithm consumes during its operation, measured in milliwatts (mW).

$$Power\ Consumption\ (mW) = \frac{Total\ Energy\ (mJ)}{Execution\ Time\ (s)} \times 1000 \quad (21)$$

Equation 21, Power consumption (mW) is the average power used during model inference, measured in milliwatts. At the same time, execution time (s) is the total time taken to process input data or perform inference.

Table 5: Power Consumption Across Dataset Attributes

Network Features	SMEESI	DAGMM	T-YOLO-IDS	EARF-AS
TCP_Flags	42	103	87	74
Src_Port	44	107	89	75
Dst_Port	43	104	88	76
Packet_Size	41	108	86	73
Flow_Duration	43	106	87	74
IoT Features	SMEESI	DAGMM	T-YOLO-IDS	EARF-AS
Device_Type	40	102	85	72
Protocol	42	105	86	73
Attack_Category	43	107	87	74
Payload_Size	41	103	85	72
Packet_Rate	42	106	86	73

SMEESI, DAGMM, T-YOLO-IDS, and EARF-AS intrusion detection algorithms are compared by power consumption in Table 5. Data is grouped by network and IoT properties. SMEESI uses the least power for all features, averaging 42.6 mW for the network and 41.6 mW for IoT. This is owing to its TinyML design, quantized convolutional neural networks, optimized autoencoder components, and fuzzy logic-controlled adaptive power management module. DAGMM uses the most power, averaging 105.6 mW for the network and 104.6 mW for the IoT. The complicated deep autoencoding and Gaussian mixture modelling components may explain its great detection accuracy, but high energy cost. T-YOLO-IDS optimally detects objects and uses modest power, balancing detection capabilities and energy needs. EARF-AS reduces computing burden with energy-aware random forests and adaptive sampling to achieve the second-best power efficiency. SMEESI reduces power usage by 60% compared to DAGMM, 51% compared to T-YOLO-IDS, and 43% compared to EARF-AS, making it appropriate for resource-constrained IoT scenarios.

4.8. Energy Consumption (mJ)

The term energy consumption refers to the total energy consumed during model execution, measured in millijoules (mJ).

$$Energy\ Consumption\ (mJ) = Power\ Consumption\ (mW) \times Execution\ Time\ (s)/1000 \quad (22)$$

Equation 22 calculates the total energy a device or process uses during its execution. It represents power consumption (mW) and execution time (s), measured in milliwatts and seconds. The result is converted from milliwatt-seconds (mWs) to millijoules (mJ) by dividing by 1000. This is done to convert the power consumed to millijoules.

Table 6: Energy consumption comparison across dataset attributes

Network Features	SMEESI	DAGMM	T-YOLO-IDS	EARF-AS
TCP_Flags	210	512	435	370
Src_Port	220	530	445	375
Dst_Port	215	520	440	375
Packet_Size	208	540	430	365
Flow_Duration	215	525	435	370
IoT Features	SMEESI	DAGMM	T-YOLO-IDS	EARF-AS
Device_Type	205	505	425	360
Protocol	210	515	430	365
Attack_Category	215	525	435	370
Payload_Size	208	510	425	360
Packet_Rate	210	520	430	365

In Table 6, the Total electrical energy consumed during algorithm execution in IoT security applications is measured in millijoules (mJ). This statistic, determined by multiplying power consumption by execution time, is a key indicator of operational sustainability for energy-constrained IoT intrusion detection systems. SMEESI's energy efficiency is high for both network (TCP_Flags, Src_Port, Dst_Port, Packet_Size, Flow_Duration) and IoT-specific features. SMEESI consumes only 213.6 mJ for network characteristics, exceeding DAGMM (525.4 mJ), T-YOLO-IDS (437 mJ), and EARF-AS (371 mJ). SMEESI uses 209.6 mJ for IoT features, compared to 364–515 mJ for competitors. SMEESI reduces energy by 60% compared to DAGMM, 51% compared to T-YOLO-IDS, and 43% compared to EARF-AS. The novel TinyML architecture, model quantization, improved neural network components, and adaptive power management system with fuzzy logic control make SMEESI highly efficient. These findings validate SMEESI's potential as a sustainable security solution for resource-constrained IoT contexts where battery longevity and energy conservation are crucial for robust cybersecurity.

4.9. Latency (ms)

Latency is the average time delay between input and system response, particularly in Edge-IIoT intrusion detection systems. Lower Latency indicates faster response, crucial for real-time or near-real-time applications where threats must be detected and mitigated promptly.

$$Latency_{avg} = \frac{1}{N}\sum_{i=1}^{N}\left(T_{response}^{(i)} - T_{input}^{(i)}\right) \qquad (23)$$

In equation 23, N is the Number of processed instances (e.g., packets, flows, or events),$T_{input}^{(i)}$ is the Timestamp when the ith input was received, $T_{response}^{(i)}$ Timestamp when the system provided a response for the ith input,$Latency_{avg}$ is the Mean Latency across all instances (measured in milliseconds, ms)

Table 7: Latency Comparison Across Dataset Attributes

Network Features	SMEESI	DAGMM	T-YOLO-IDS	EARF-AS
TCP_Flags	12.1	33.7	28.4	23.5
Src_Port	12.5	34.1	29.0	24.0
Dst_Port	12.3	33.9	28.7	23.8
Packet_Size	12.0	34.2	28.2	23.4
Flow_Duration	12.4	34.0	28.6	23.7
IoT Features	SMEESI	DAGMM	T-YOLO-IDS	EARF-AS
Device_Type	11.8	33.5	28.0	23.3
Protocol	12.2	33.8	28.5	23.6
Attack_Category	12.6	34.0	28.8	23.9
Payload_Size	12.0	33.6	28.1	23.4
Packet_Rate	12.3	33.7	28.4	23.6

Table 7 shows that Latency, measured in milliseconds, is a crucial performance indicator in Edge-IIoT intrusion detection systems. It directly impacts the operational effectiveness of security mechanisms, especially in scenarios requiring immediate threat identification and mitigation. SMEESI outperforms four studied algorithms in network and IoT feature processing domains, with low latency values ranging from 12.0 to 12.5 ms for network attributes and 11.8 to 12.6 ms for IoT-specific features. SMEESI achieves an average reduction of 64% in response time compared to DAGMM, 57% compared to T-YOLO-IDS, and 48% compared to EARF-AS. This exceptional processing speed is attributed to SMEESI's lightweight quantized neural network architecture, efficient computational pipeline design, and optimized implementation for resource-constrained edge devices. This reduced Latency ensures real-time protection for time-critical IoT applications in domains like industrial control systems, healthcare monitoring, and smart infrastructure, where delayed threat detection could lead to severe consequences.

5 Conclusion and Future Work

The research has developed and validated SMEESI, a TinyML-enabled intrusion detection system that addresses the challenge of implementing robust security in energy-constrained IoT environments. The experimental results using the EdgeIIoTset dataset demonstrate that SMEESI achieves a significant reduction in resource consumption while maintaining high detection accuracy, thus providing a sustainable security solution for IoT ecosystems. Key achievements include developing quantized neural architectures that operate effectively within the severe memory constraints of IoT devices, achieving 94.3% detection accuracy while requiring only 4.2 kB of memory; implementing an adaptive power management framework using Fuzzy Logic Control that intelligently adjusts computational intensity based on threat levels, reducing energy consumption by 78% compared to conventional IDS approaches; and creating a multi-level detection pipeline that extends device operational lifetime by 3.6× while maintaining security vigilance against diverse attack vectors present in the EdgeIIoTset dataset. The practical implications of this research are substantial for advancing green cybersecurity practices. By enabling effective security monitoring with minimal energy consumption, SMEESI helps extend the operational lifetime of IoT devices, reducing the frequency of battery replacements or recharges, and contributing to sustainability through reduced electronic waste generation and lower maintenance requirements. Field testing across smart buildings, industrial monitoring, and healthcare applications has validated the system's versatility and effectiveness in real-world scenarios, demonstrating the practical feasibility of sustainable security approaches even in resource-constrained environments.

References

[1] Abdulganiyu, O. H., Tchakoucht, T. A., & Saheed, Y. K. (2024). Retracted Article: Towards an efficient model for network intrusion detection system (IDS): systematic literature review. *Wireless networks*, *30*(1), 453-482. https://doi.org/10.1007/s11276-023-03495-2

[2] Ahmed, S. T., Kumar, V. V., & Jeong, J. (2024). Heterogeneous workload-based consumer resource recommendation model for smart cities: EHealth edge–cloud connectivity using federated split learning. *IEEE Transactions on Consumer Electronics*, *70*(1), 4187-4196. https://doi.org/10.1109/TCE.2024.3374462

[3] Alsulami, M. H. (2024). An AI-driven model to enhance sustainability for the detection of cyber threats in IoT environments. *Sensors*, *24*(22), 7179. https://doi.org/10.3390/s24227179

[4] Ball, C. S., & Degischer, D. (2024). IoT implementation for energy system sustainability: The role of actors and related challenges. *Utilities Policy*, *90*, 101769. https://doi.org/10.1016/j.jup.2024.101769

[5] Canavese, D., Mannella, L., Regano, L., & Basile, C. (2024). Security at the edge for resource-limited IoT devices. *Sensors*, *24*(2), 590. https://doi.org/10.3390/s24020590

[6] Cano-Suñén, E., Martínez, I., Fernández, Á., Zalba, B., & Casas, R. (2023). Internet of Things (IoT) in buildings: A learning factory. *Sustainability*, *15*(16), 12219. https://doi.org/10.3390/su151612219

[7] Chatterjee, R., & Chakraborty, R. (2024). Lightweight Cryptography: Methods and Systems for Securing Resource-Constrained Devices. *International Journal of Communication*, *1*(1), 1-22.

[8] Chlaihawi, M. O. A. (2024). Application of Blockchain Technology to Reduce Costs. *International Academic Journal of Social Sciences*, *11*(1), 26-38.

[9] Escobedo, F., Canales, H. B. G., Galarza, F. W. M., Saldaña, C. M. A., Reyes, E. M. A., & Tananta, C. A. F. (2024). Energy Efficient Business Management System for Improving QoS in Network Model. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *15*(1), 42-52. https://doi.org/10.58346/JOWUA. 2024.I1.004

[10] Ferrag, M. A. (2022). Edge-IIoTset cyber security dataset of IoT & IIoT. https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot

[11] Hashemi, M. S. (2016). The effect of infrastructure, corporate culture, organizational structure and information technology on Competitive Intelligence in Organizations. *Human Resource Management*, *3*(3), 43-50. https://doi.org/10.9756/IAJOBHRM/V6I1/1910003

[12] Hussain, I., & Qureshi, A. (2024). Gender-Inclusive Energy Transitions: Empowering Women in Renewable Energy Sectors. *International Journal of SDG's Prospects and Breakthroughs*, *2*(2), 7-9.

[13] Ige, A. B., Kupa, E., & Ilori, O. (2024). Best practices in cybersecurity for green building management systems: Protecting sustainable infrastructure from cyber threats. *International Journal of Science and Research Archive*, *12*(1), 2960-2977. https://doi.org/10.30574/ijsra.2024.12.1.1185

[14] Kallimani, R., Pai, K., Raghuwanshi, P., Iyer, S., & López, O. L. (2024). TinyML: Tools, applications, challenges, and future research directions. *Multimedia Tools and Applications*, *83*(10), 29015-29045. https://doi.org/10.1007/s11042-023-16740-9

[15] Katib, I., Albassam, E., Sharaf, S. A., & Ragab, M. (2025). Safeguarding IoT consumer devices: Deep learning with TinyML driven real-time anomaly detection for predictive maintenance. *Ain Shams Engineering Journal*, *16*(2), 103281. https://doi.org/10.1016/j.asej.2025.103281

[16] Krause, T., Ernst, R., Klaer, B., Hacker, I., & Henze, M. (2021). Cybersecurity in power grids: Challenges and opportunities. *Sensors*, *21*(18), 6225. https://doi.org/10.3390/s21186225

[17] Krishnan, V. G., Krishnan, T. N., Karim, S. S., Yuvarajan, G., & Priya, M. R. (2020). Cyber Security in Data Mining to Data Driven Security. *International Journal of Advances in Engineering and Emerging Technology*, *11*(1), 71-76.

[18] Kulkarni, M. D., & Angurala, M. P. (2024). Trends in Computer Science and Information Technology Research. *International Journal of Human–Computer Interaction*, *36*(6).

[19] Morchid, A., Jebabra, R., Ismail, A., Khalid, H. M., El Alami, R., Qjidaa, H., & Jamil, M. O. (2024). IoT-enabled fire detection for sustainable agriculture: A real-time system using flask and embedded technologies. *Results in Engineering*, *23*, 102705. https://doi.org/10.1016/j.rineng.2024.102705

[20] Oliveira, F., Costa, D. G., Assis, F., & Silva, I. (2024). Internet of Intelligent Things: A convergence of embedded systems, edge computing and machine learning. *Internet of Things*, *26*, 101153. https://doi.org/10.1016/j.iot.2024.101153

[21] Pandey, S., & Bhushan, B. (2024). Recent Lightweight cryptography (LWC) based security advances for resource-constrained IoT networks. *Wireless Networks*, *30*(4), 2987-3026.

[22] Ranpara, R., Alsalman, O., Kumar, O. P., & Patel, S. K. (2025). A simulation-driven computational framework for adaptive energy-efficient optimization in machine learning-based intrusion detection systems. *Scientific Reports*, *15*(1), 13376. https://doi.org/10.1038/s41598-025-93254-4

[23] Rekeraho, A., Cotfas, D. T., Balan, T. C., Cotfas, P. A., Acheampong, R., & Tuyishime, E. (2025). Cybersecurity Threat Modeling for IoT-Integrated Smart Solar Energy Systems: Strengthening Resilience for Global Energy Sustainability. *Sustainability*, *17*(6), 2386. https://doi.org/10.3390/su17062386

[24] Rekeraho, A., Cotfas, D. T., Cotfas, P. A., Bălan, T. C., Tuyishime, E., & Acheampong, R. (2024). Cybersecurity challenges in IoT-based smart renewable energy. *International Journal of Information Security*, *23*(1), 101-117.

[25] Rishikesh, R., Tamilselvan, Y., & Sujai, S. (2022). Intrusion of Attacks in Puppet and Zombie Attacking and Defence Model Using BW-DDOS. *International Academic Journal of Innovative Research*, *9*(1), 13-19.

[26] Rupanetti, D., & Kaabouch, N. (2024). Combining edge computing-assisted internet of things security with artificial intelligence: Applications, challenges, and opportunities. *Applied Sciences*, *14*(16), 7104. https://doi.org/10.3390/app14167104

[27] Shetty, A., & Nair, K. (2024). Artificial Intelligence Driven Energy Platforms in Mechanical Engineering. *Association Journal of Interdisciplinary Technics in Engineering Mechanics*, *2*(1), 23-30.

[28] Suryavanshi, N. S., Acharya, P. S., & Jadhav, A. N. (2025). A Trust-Security-Resource Optimization Framework for Cloud-Edge-IoT Collaboration in Industrial Applications. *IJSAT-International Journal on Science and Technology*, *16*(1). https://doi.org/10.71097/IJSAT.v16.i1.1626

[29] Tan, W., Sarmiento, J., & Rosales, C. A. (2024). Exploring the Performance Impact of Neural Network Optimization on Energy Analysis of Biosensor. *Natural and Engineering Sciences*, *9*(2), 164-183. https://doi.org/10.28978/nesciences.1569280

[30] Tekin, N., Acar, A., Aris, A., Uluagac, A. S., & Gungor, V. C. (2023). Energy consumption of on-device machine learning models for IoT intrusion detection. *Internet of Things*, *21*, 100670.

[31] Wang, Y., & Liu, L. (2024). Research on sustainable green building space design model integrating IoT technology. *PloS one*, *19*(4), e0298982. https://doi.org/10.1371/journal.pone.0298982

[32] Zhukabayeva, T., Zholshiyeva, L., Karabayev, N., Khan, S., & Alnazzawi, N. (2025). Cybersecurity solutions for industrial internet of things–edge computing integration: Challenges, threats, and future directions. *Sensors*, *25*(1), 213. https://doi.org/10.3390/s25010213

Authors Biography

Rajalakshmi Selvaraj is an associate professor in the Department of Computing and Informatics at the Botswana International University of Science and Technology (BIUST). She earned her doctorate in network security with a focus on honeypots from the University of Johannesburg, South Africa. She brings over 15 years of experience collaborating with various companies and industry sectors on student projects. Throughout her career, she has been deeply involved in teaching, research, and strategic initiatives. She is passionate about mentoring the next generation of global design leaders. Currently, she is working on a project to develop a security system for honeypot architecture aimed at preventing attacks on honeypots. Selvaraj is an active member of numerous research-promoting committees, including IEEE, ACM, and CSSA. She has published over 60 articles in accredited journals, conference proceedings, book chapters, and textbooks. Additionally, she holds four international patents and has supervised a significant number of MSc and Ph.D. students.

Venu Madhav Kuthadi holds a bachelor's degree in computer science and engineering from Nagarjuna University, India, and a master's degree in computer science from JNT University, India, completed in 2001. He earned his doctorate in engineering from the University of Johannes Burg in 2018. Kuthadi served as a senior lecturer in the Department of Applied Information Systems at the University of Johannesburg from March 2000 to January 2017. He is currently an associate professor in the Department of Computing and Informatics at the Botswana International University of Science and Technology (BIUST). His research focuses on network security, specifically in developing security patterns to protect data transmitted over networks. He introduced an adaptive pre-processing technique using principal component analysis (PCA) and hyperbolic Hopfield neural network (HHNN) to enhance the efficiency of streaming data. He has an extensive publication record with over 50 peer-reviewed journal articles, two textbooks, and more than 20 international conference proceedings. He has successfully supervised 10 master's students and 3 Ph.D. candidates. Additionally, he serves as an editor for the IJAEGT journal and is a reviewer for several reputable journals.

Dr.S. Baskar, Assistant Professor in the Department of Electronics and Communication Engineering/Center for Interdisciplinary Research, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India. He worked as a Research Associate in the departments of Nano Science and Technology and Pervasive Computing Technology. His research interests include Wireless sensor networks, Low power VLSI, IoT, Material Science. He published more than 33 research articles in international journals, book chapters and Conferences. So far, he guided 20 M.E/M. Tech/B. E projects. He is a reviewer for referred journals like wireless sensor networks, IEEE Access, Inder science 5and IEEE transactions on vehicular technology. He is a Young Scientist Awardee by the Department of Science and Technology, Government of India.

Roberto Acevedo is a faculty member at the Facultad de Ingeniería, Universidad San Sebastián, Santiago, Chile. Based at the Bellavista 7 campus, he brings extensive academic and practical experience in the field of engineering, with particular interests in systems engineering, automation, and emerging digital technologies. Roberto's research contributions focus on applying engineering principles to solve real-world problems, including sustainable technology solutions and process optimization. He has presented his work at national and international forums and contributes to peer-reviewed publications. Roberto is also involved in outreach initiatives aimed at promoting STEM education in Chile. His dedication to academic excellence and student-centered learning makes him a valued member of the engineering faculty at Universidad San Sebastián.