A PRACTICAL LOW-COST NILM DEVICE BASED ON TINY MACHINE LEARNING

Bien Nguyen Quang[*], Long Hoang Thanh[+], Loi Tran Van[+], Duan Luong Cong[*]
[*] Faculty of Electronics Engineering 1, Posts and Telecommunications Institute of Technology
[+] Faculty of Telecommunication Engineering 1, Posts and Telecommunications Institute of Technology

***Abstract*:** This study addresses the growing importance of Non-Intrusive Load Monitoring (NILM) in enhancing energy efficiency in load consumption monitoring. The objective of this research is to develop an integrated system that utilizes NILM, combining TinyML and IoT technologies, for real-time monitoring and control of household devices. This approach leverages the efficiency of TinyML for on-device processing while enabling seamless connectivity and data management through IoT. We employed a Random Forest machine learning model alongside the ESP32 MCU to achieve this goal. Key findings indicate that the system can classify various load types with high accuracy and minimal latency, demonstrating effective performance in realworld conditions. The implications of this study suggest that NILM can significantly improve user engagement in energy management while offering a cost-effective solution for load consumption monitoring.

Keywords: Load Monitoring, Load Disaggregation, Nonintrusive Load Monitoring, NILM, TinyML, Random Forest.

I. INTRODUCTION.

Non-Intrusive Load Monitoring (NILM) is an innovative method that analyzes the aggregate electrical load of a building and disaggregates it into individual appliances, allowing for energy consumption tracking without dedicated sensors on each device [1]. By leveraging data from a single whole-home electricity meter, NILM systems provide insights into energy usage patterns, appliance efficiency, and conservation opportunities [2], [3]. Unlike traditional utility meters that offer only total consumption, NILM delivers detailed, actionable information at the appliance level, enabling users to achieve electricity savings of approximately 12% through continuous feedback, compared to 3.8% from monthly total consumption feedback. As demand for energy-efficient buildings increases, NILM presents a cost-effective and scalable solution for enhancing energy visibility and informed decision-making [5].

The term "non-intrusive" indicates minimal disruption to user privacy, as measurements are taken from a single aggregated load point, avoiding the need for additional equipment that increases installation complexity and cost. Each electrical appliance has unique energy consumption patterns known as "load signatures," allowing for categorization into several types: ON/OFF state appliances (e.g., lamps and toasters), Finite State Machine (FSM) appliances (e.g., washing machines), Continuously Variable Devices (CVD) (e.g., dimmer lights), and permanent consumer devices (e.g., TV receivers). These categories reflect the varying operational states and complexities in disaggregating total consumption [6].

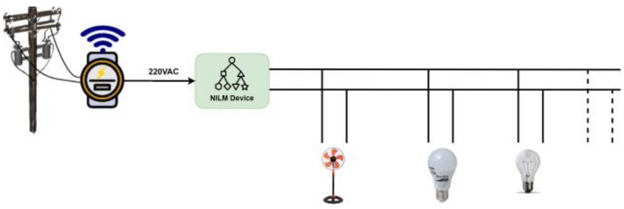

Figure 1. NILM device on power line system.

NILM algorithms effectively infer appliance usage from a single metering point, typically using machine learning to detect ON/OFF events. However, their high computational demands often necessitate cloud services for appliance-level energy feedback, raising consumer privacy concerns [7], [8]. Additionally, existing commercial smart meters typically have sampling frequencies with reporting intervals of 15 minutes to daily, while NILM algorithms require a minimum frequency of 1 Hz [9]. Consequently, despite the prevalence of smart meters, leveraging NILM for detailed energy feedback in residential settings remains challenging.

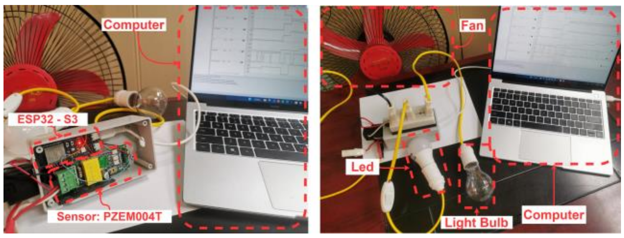

Figure 2. Data collection device and data acquisition.

A promising solution to overcome these limitations is the use of TinyML - a branch of ML focused on deploying small-sized, high-performance, and energyefficient ML models [10]. TinyML algorithms can be deployed directly on embedded processors like MCUs, allowing on-device data processing without the need for cloud connectivity

Contact author: Duan Luong Cong
Email: duanlc@ptit.edu.vn
Manuscript received: 31/10/2024, revised: 26/11/2024, accepted: 09/12/2024.

[11], [12]. This helps address the computational resource issue and improves data privacy for users. Integrating TinyML into smart metering devices can provide detailed insights into the energy consumption of individual electronic appliances within a household, thereby helping users manage their energy consumption more effectively [13], [14].

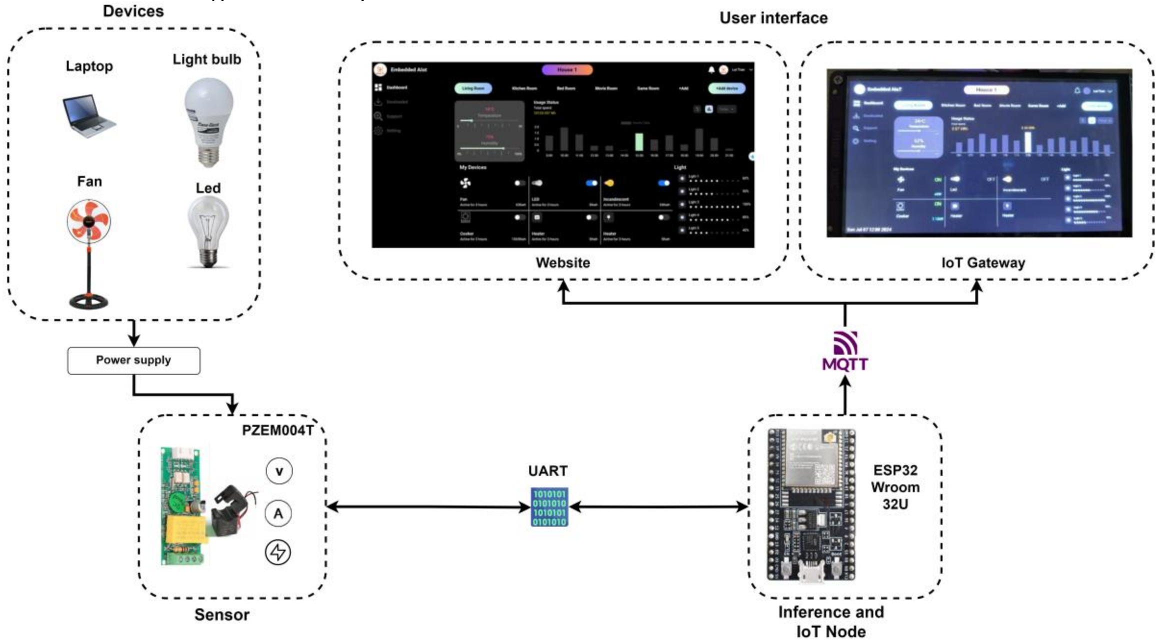

Figure 3. System implementation using edge classification.

Recent research has focused on developing embedded NILM systems for household energy management, utilizing heterogeneous computing to perform on-site load disaggregation and transforming power meters into "cognitive power meters" [15]. Low-cost embedded NILM systems with low sampling rates have been proposed, achieving high accuracy in identifying various household appliances [16]. These systems extract features such as active and reactive power changes and use machine learning techniques like k-Nearest Neighbors for appliance classification [17], [15]. They can detect over 90% of total events and accurately disaggregate devices and measure total power consumption [16]. By providing insights into appliance usage, these systems aid in energy conservation and enhance power management in domestic and industrial settings.

In this study, we developed a system that combines TinyML and IoT, enabling users to monitor the operational status of load devices through an affordable embedded device. The results of this research present a small-scale machine learning model capable of distinguishing the activities of various loads based on current, power, and power factor. Additionally, we successfully built a prototype system capable of operating under real-world conditions.

The structure of this paper is organized as follows: Section 2 details the methodology, including data collection, pre-processing, model training, and implementation procedures. Section 3 presents the results along with an in-depth discussion of the findings.

II. METHODOLOGY

A. Hardware, Datasets Collecting & Pre-Processing

To collect data, we developed a device capable of measuring specific system parameters, as illustrated in Figure 2-a. This device is built around the ESP32 MCU, which supports Wi-Fi connectivity for efficient data acquisition. It features floating-point unit (FPU) and digital signal processing (DSP) capabilities,

allowing it to execute machine learning models, as detailed in Section II-C. The ESP32 is connected to the PZEM004T module to gather electrical parameters, providing information on voltage (U), current (I), power (P), and power factor (cos) of the load, with a maximum sampling frequency of 10 Hz. We developed firmware for the ESP32, enabling it to read parameters and transmit data to a PC via the MQTT protocol. Data from each 10-second window, containing 100 entries for three signal fields, resulting in a total of 300 values, is aggregated and sent to the PC periodically. Additionally, a software application written in Python was created to store the received data in a CSV file. The data collection process is illustrated in Figure 2-b.

For datasets creation, we utilized 6 load devices: (D1) incandescent bulb (ON/OFF), (D2) fan (FSM), (D3) LED light (ON/OFF), (D4) laptop (CVD), (D5) television (CVD), and (D6) induction cook-top (FSM). The devices were sequentially connected to the power supply, allowing for combinations of 1, 2, 3, and 6 devices in full arrangements. Modes with 4 and 5 devices were implemented randomly in certain instances. For each configuration change, data was recorded at least three minutes before transitioning to the next mode. The final datasets obtained encompass 5.2 hours of data.

Before testing the machine learning model, the data underwent pre-processing to create segments of 1 second, each containing three key parameters: current (I), power (P), and power factor (cos). The segmentation process prioritized segments with consistent load labels, while segments with differing labels were excluded. Following this pre-processing step, the datasets contained a total of 37,604 samples available for training and testing. The input data for the model comprised 30 features, providing a comprehensive representation of the measured parameters.

B. Model Design

In this model, we employed Random Forest (RF) [18] to identify the activity of load devices. Previous studies have demonstrated that RF performs effectively in NILM applications with low sampling frequencies [19], [20], [21]. In addition to optimizing accuracy, we also conducted experiments to optimize the number of decision trees in order to balance accuracy with resource utilization and execution time, enabling effective optimization and implementation on the MCU. The dataset, comprising a total of 37,604 samples, was divided into two main parts: 80% was allocated for training *(with 60% used for training and 20% for validation)*, while the remaining 20% was designated for testing.

C. Implementation

After training, the RF model was converted into C++ source code using the MicroML1 tool. This generated code was utilized to develop the system code and model, as illustrated in Figure 3. The device firmware reads data from the PZEM004T module via UART communication. The obtained sensor data is normalized and input into the model for classification. Based on the analysis results, the device statuses are updated on the gateway through MQTT communication.

Additionally, we developed an IoT gateway to display the status of household devices and enable their control. This gateway is built on the IDO-SMLCD72- V1-2EC circuit board, operating on the OpenWrt2 platform. A web platform was also created for remote monitoring and control of the system over the Internet. All devices and components communicate with each other via MQTT connections, allowing the synchronization of device statuses and supporting remote monitoring.

III. RESULTS AND DISCUSSION

A. Model performance

Based on the methodology outlined in Section II-B, we systematically varied the number of estimators in the Random Forest model from 2 to 32 to evaluate accuracy, precision, recall, and F1-score. Additionally, we integrated these models into the ESP32 platform to assess flash size and execution time. The results are summarized in Table I.

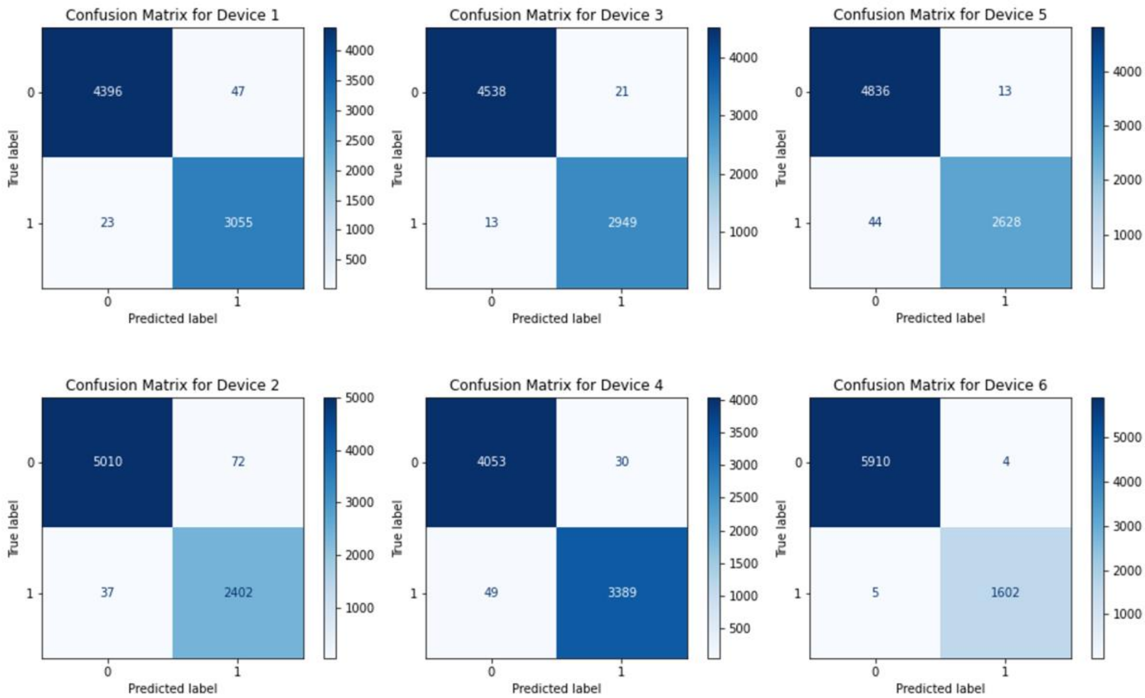

Figure 4. Confusion matrix on test set over 6 devices.

Our findings demonstrate that as the number of estimators increases from 2 to 6, the model's accuracy approaches 97%. For estimators ranging from 7 to 16, the model achieves precision, recall, and F1-score values of 98%, 0.97, 0.97, and 0.97, respectively, while maintaining a model size under 200KB and an execution time of less than 157ms. However, when increasing the number of estimators to 32, we observed only marginal improvements in accuracy, accompanied by a significant increase in model size and execution time (310ms). Consequently, we selected 7 estimators for further integration and evaluation in real-world applications. This configuration utilizes only 81.3KB of flash memory and requires 71ms of CPU time on the ESP32-S3 for inference.

The model was subsequently utilized to assess its ability to distinguish between 6 different devices. The results revealed that it performed effectively in identifying high-power loads, such as D6 and D5, achieving high accuracy levels. In contrast, the accuracy for the other devices showed a slight decrease, with the fan device exhibiting the lowest accuracy at around 97%. Detailed results are presented in Figure 4, where Label 1 indicates a device identified as active and Label 0 signifies a device that is inactive. Overall, these findings indicate that the model maintains relatively consistent performance across various load types, operational modes, and power ranges, underscoring its reliability for practical applications.

Table 1

EXECUTION RESULTS OF THE MODEL OVER NUMBER ESTIMATORS

Estimator	Accuracy	Precision	Recall	F1-score	Flash (KB)	Execution (ms)
2	0.95	0.97	0.95	0.96	23.1	22.9
3	0.96	0.97	0.97	0.97	35.1	32.7
4	0.97	0.97	0.97	0.97	47.7	42.3
5	0.97	0.97	0.97	0.97	56.8	51.7
6	0.97	0.97	0.97	0.97	70.1	61.6
7	**0.98**	**0.97**	**0.97**	**0.97**	**81.3**	**71.2**
8	0.98	0.97	0.97	0.97	92.9	80.8
9	0.98	0.97	0.97	0.97	103.7	90.3
10	0.98	0.97	0.97	0.97	115.2	99.8
11	0.98	0.97	0.97	0.97	126.5	109.5
12	0.98	0.97	0.97	0.97	136.8	118.8
16	0.98	0.97	0.97	0.97	181.2	157.3
32	0.98	0.98	0.98	0.97	365.6	310.9

B. *System evaluation*

In this section, we evaluate the overall performance of the integrated system, focusing on its operational efficiency and responsiveness in real-world scenarios. Building on the selected machine learning model, we developed firmware for the ESP32-S3 as outlined in Section II-C. In addition to its capability to read power data and classify active loads, the device can synchronize its identification status with the gateway and the monitoring web interface, achieving an update frequency of once per minute. Furthermore, the device is designed to receive control commands from users through two interfaces, allowing for the toggling of devices connected to the corresponding relay positions. This functionality enhances user interaction and ensures effective management of household devices.

In addition, we successfully developed the gateway and web monitoring interface. These user interfaces enable real-time and historical monitoring of each device’s operational status. We also implemented energy consumption monitoring features, allowing users to track power usage across different time intervals throughout the day, thereby enhancing energy management in household settings.

Finally, we conducted tests of the system under realworld conditions. The model of the device used for the testing is illustrated in Figure 5. The results of the real-world testing demonstrate that the model can classify, synchronize status, and control devices effectively, achieving real-time response with a maximum classification delay of 1 minute following a change in load status. A demo of the real-world testing process can be found at the following link3.

The results of this study highlight the applicability of NILM in real-world conditions, particularly in the domains of home monitoring and smart home systems. Furthermore, the potential of this research lies in its ability to be deployed at a low cost while maintaining suitable accuracy, thereby enhancing user experience in the increasingly evolving landscape of IoT and smart home technologies.

IV. CONCLUSIONS

This study demonstrates the successful integration of TinyML and IoT technologies within NILM for efficient electrical load monitoring. Using a Random Forest (RF) machine learning model, we achieved high classification accuracy of approximately 98%, with an execution time of just 71 milliseconds and a maximum classification delay of one minute. These results underscore NILM’s potential to enhance user experiences in smart homes through a cost-effective solution. The integration of TinyML and IoT not only improves load classification but also facilitates seamless data management and control. Future work will focus on researching algorithms capable of processing higherfrequency data to improve accuracy and enhance the diversity of datasets.

ACKNOWLEDGMENT

This research is sponsored by the Posts and Telecommunications Institute of Technology under project code 11-2024-HV-KTDT1.

REFERENCES

[1] G. Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[2] M. Zeifman and K. Roth, “Nonintrusive appliance load monitoring: Review and outlook,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, Feb. 2011.

[3] J. Alcalá, J. Urena, Á. Hernández, and D. Gualda, “Assessing ˜ Human Activity in Elderly People Using Non-Intrusive Load Monitoring,” *Sensors*, vol. 17, no. 2, p. 351, Feb. 2017.

[4] M. A. Alahmad, P. G. Wheeler, A. Schwer, J. Eiden, and A. Brumbaugh, “A Comparative Study of Three Feedback Devices for Residential Real-Time Energy Monitoring,” IEEE Transactions on Industrial Electronics, vol. 59, no. 4, pp. 2002– 2013, Apr. 2012.

[5] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, “Enhancing Electricity Audits in Residential Buildings with Nonintrusive Load Monitoring,” Journal of Industrial Ecology, vol. 14, no. 5, pp. 844–858, Oct. 2010.

[6] G.-F. Angelis, C. Timplalexis, S. Krinidis, D. Ioannidis, and D. Tzovaras, “NILM applications: Literature review of learning approaches, recent developments and challenges,” Energy and Buildings, vol. 261, p. 111951, Apr. 2022.

[7] S. Steephen, S. R, and N. N, “A Review on Non-Intrusive Load Monitoring Using Deep Learning,” in 2022 International Conference on Innovations in Science and Technology for Sustainable Development (ICISTSD). Kollam, India: IEEE, Aug. 2022, pp. 381–385.

[8] L. W. Tokam and S. S. Ouro-Djobo, “Non-Intrusive Load Monitoring (NILM), Interests and Applications,” Journal of the Nigerian Society of Physical Sciences, p. 1208, May 2023.

[9] M. Zeifman, “Disaggregation of home energy display data using probabilistic approach,” IEEE Transactions on Consumer Electronics, vol. 58, no. 1, pp. 23–31, Feb. 2012.

[10] W. Pete and D. Situnayake, TinyML Machine Learning with TensorFlow on Arduino and Ultra-Low Power MicroControllers, 2020.

[11] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, “TinyML: Tools, applications, challenges, and future research directions,” Multimedia Tools and Applications, vol. 83, no. 10, pp. 29 015–29 045, Sep. 2023.

[12] H. Oufettoul, R. Chaibi, and S. Motahhir, “TinyML Applications, Research Challenges, and Future Research Directions,” in 2024 21st Learning and Technology Conference (L&T). Jeddah, Saudi Arabia: IEEE, Jan. 2024, pp. 86–91.

[13] Z. Wu, C. Wang, L. Xiong, R. Li, T. Wu, and H. Zhang, “A Smart Socket for Real-Time Nonintrusive Load Monitoring,” IEEE Transactions on Industrial Electronics, vol. 70, no. 10, pp. 10 618–10 627, 2023-10.

[14] A. Gozuoglu, O. Ozgonenel, and C. Gezegin, “CNN-LSTM based deep learning application on Jetson Nano: Estimating electrical energy consumption for future smart homes,” Internet of Things, vol. 26, p. 101148, 2024-07.

[15] F. Deluno Garcia, W. Angelino De Souza, and F. Pinhabel Marafão, “Embedded NILM as Home Energy Management System: A Heterogeneous Computing Approach,” IEEE Latin America Transactions, vol. 18, no. 02, pp. 360–367, Feb. 2020.

[16] S. Biansoongnern and B. Plangklang, “An Alternative LowCost Embedded NILM System for Household Energy Conservation with a Low Sampling Rate,” Symmetry, vol. 14, no. 2, p. 279, Jan. 2022.

[17] G. Braglia and A. E. Lazzaretti, “An Embedded System for NILM Using Machine Learning,” in Anais Do 15. Congresso Brasileiro de Inteligência Computacional. SBIC, Jan. 2021, pp. 1–8.

[18] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[19] Y. Yan, L. Zhu, A. Yang, Y. Zhi, B. Liu, Y. Huang, and H. Wen, “A Review of Industrial Non-Intrusive Load Monitoring,” in 2023 International Conference on Sensing, Measurement & Data Analytics in the Era of Artificial Intelligence (ICSMD). IEEE, 2023-11-02, pp. 1–6.

[20] P. A. Schirmer and I. Mporas, “Non-Intrusive Load Monitoring: A Review,” IEEE Transactions on Smart Grid, vol. 14, no. 1, pp. 769–784, 2023-01.

[21] S. Sykiotis, S. Athanasoulias, M. Kaselimi, A. Doulamis, N. Doulamis, L. Stankovic, and V. Stankovic, “PerformanceAware NILM Model Optimization for Edge Deployment,” IEEE Transactions on Green Communications and Networking, vol. 7, no. 3, pp. 1434–1446, Sep. 2023.

THIẾT BỊ GIÁM SÁT TẢI KHÔNG XÂM NHẬP (NILM) CHI PHÍ THẤP DỰA TRÊN HỌC MÁY CỠ NHỎ (TINYML)

Tóm tắt: Nghiên cứu này triển khai thiết bị giám sát tải không xâm nhập (NILM) để theo dõi tiêu thụ điện năng, kết hợp công nghệ IoT và TinyML. Thiết bị được xây dựng và thử nghiệm trên tập dữ liệu gồm sáu loại thiết bị điện gia dụng phổ biến. Dựa trên dữ liệu thu thập, nghiên cứu phát triển giải pháp phân loại tải NILM tại biên với kết nối IoT, cho phép giám sát và điều khiển từ xa. Vi điều khiển ESP32 được chọn để tích hợp mô hình học máy rừng ngẫu nhiên, phân loại tải dựa trên các tham số công suất, dòng điện và hệ số công suất, với thời gian phân loại 71ms. Kết quả thử nghiệm cho thấy hệ thống có khả năng phân loại nhiều loại tải với độ chính xác cao trong thời gian thực. Nghiên cứu nhấn mạnh vai trò của TinyML trong phân loại tải không xâm nhập, đảm bảo sự cân bằng giữa hiệu năng và độ chính xác.

Từ khóa: Giám sát tải, Phân tách tải, Giám sát tải không xâm nhập, NILM, Học máy nhỏ, Rừng ngẫu nhiên.

Bien Nguyen Quang, received a BE degree in Electrical - Electronic Engineering in 2022 and an MSc in Electronic Engineering in 2024 from the Posts and Telecommunications Institute of Technology (PTIT). He is a lecturer in the Faculty of Electronic Engineering 1 at PTIT. His main research areas include Embedded Systems, Internet of Things (IoT), FPGA, and Tiny Machine Learning

Email: biennq@ptit.edu.vn

Long Hoang Thanh, is a student of the D19 cohort majoring in Telecommunications Electronics at Posts and Telecommunications Institute of Technology (PTIT). He is currently an intern at VNPT Technology.

Email: longht.b20vt233@stu.ptit.edu.vn

Loi Tran Van, is a student of the D20 cohort majoring in Telecommunications Electronics at Posts and Telecommunications Institute of Technology (PTIT).

Email:
loitv.b21vt273@stu.ptit.edu.vn

Duan Luong Cong, received a BE degree in Electrical - Electronic Engineering in 2014 and an MSc in Telecommunication Engineering in 2018 from the Posts and Telecommunications Institute of Technology (PTIT). He is a lecturer in the Faculty of Electronic Engineering 1 at PTIT. His main research areas include Embedded Systems, DSP, Internet of Things (IoT), and TinyML.

Email:
longht.b20vt233@stu.ptit.edu.vn