

An Intrusion Detection Architecture Based on the Energy Consumption of Sensors Against Energy Depletion Attacks in LoRaWAN

André Proto¹^a, Charles C. Miers²^b and Tereza C. M. B. Carvalho¹^c

¹Laboratory of Sustainability in ICT, University of São Paulo (USP), Brazil

²Graduate Program in Applied Computing (PPGCA), Santa Catarina State University (UDESC), Brazil


Keywords: Energy Depletion Attacks, LoRaWAN, Distance Metrics, Intrusion Detection, Lightweight Architecture.


Abstract: LoRaWAN emerges as a promising technology for deploying low-power sensors to tackle industrial and urban challenges. However, Energy Depletion Attacks (EDAs) presents a substantial threat to sensors operating within the LoRaWAN framework. Various attacks, including jamming, replay attacks, firmware manipulation, and application vulnerabilities in Internet of Things systems, have the potential to induce energy depletion. Some of them are regarded as silent attacks, characterized by the absence or minimal occurrence of network traffic, rendering their detection challenging. In response to this challenge, our research introduces an architecture designed to detect EDAs in LoRaWAN sensors. We propose an implementation of a lightweight and energy-efficient intrusion detection system developed for resource-constrained devices. Our solution applies distance metrics to detect anomaly behaviours in the energy consumption patterns of sensors. In order to assess the viability of our proposed methodology, we employ the F1 score as an evaluative metric that demonstrates the efficiency of its intrusion detection accuracy of EDAs. Thus, our proposal diverges from the traditional approaches relying on network traffic analysis for intrusion detection, opting instead for a focus on the analysis of energy consumption data.


1 INTRODUCTION

The LoRaWAN protocol, designed for Low Power Wide Area Network (LPWAN) applications, facilitates the wireless connectivity of battery-operated devices within the Internet of Things (IoT) (LoRa Alliance, 2020). It finds application in diverse sectors such as smart cities, agriculture, and industry (Raza et al., 2017). Employing a star topology for communication, LPWAN enables direct communication between sensors and gateways. The LoRaWAN offers three classes of sensors: A, B, and C, with class A being the most widely utilized. In this class, communication must be initiated only by the sensor, providing advantages such as cost-effectiveness, minimal energy consumption, extended communication ranges, compatibility with heterogeneous devices, and scalability.

In the context of security, Energy Depletion Attacks (EDAs) have proven effective in disrupting services within LPWANs (Mikhaylov et al., 2019; Nguyen et al., 2019). These attacks aim to deplete sensor batteries, rendering them inoperable by exhausting their energy reserves. EDAs have the potential to impact many sensors, leading to severe damage to the overall IoT system and incurring substantial maintenance costs. Some attacks exploit network vulnerabilities to elevate sensor transmission activity; Furthermore, other types of attacks, which we call silent EDAs, exploit hardware or software vulnerabilities to increase sensor processing or internal component activity (Kuaban et al., 2023). Existing research on EDAs detection and mitigation primarily concentrates on specific EDA types, primarily analysing traffic behaviour, with a predominant focus on IoT networks like Low-power and Lossy Networks (LLNs) (Alsirhani et al., 2022;

^a <https://orcid.org/0000-0002-7250-2451>

^b <https://orcid.org/0000-0002-1976-0478>

^c <https://orcid.org/0000-0002-0821-0614>

Jan et al., 2019; Pu, 2019). LLNs employ different topologies and protocols, such as mesh topology and the Routing Protocol for LLNs (RPL). Other researchers just address mitigation strategies for specific vulnerabilities in LoRaWAN (Sciancalepore et al., 2021). Detecting zero-day and silent EDAs in LoRaWAN poses persistent challenges, particularly in developing a unified solution. Moreover, any proposed solution must be lightweight to operate efficiently on resource-constrained devices, which face limitations in processing, energy, and network resources.

We propose a lightweight architecture for detecting EDAs in LoRaWAN, which we call as LADE. This architecture has been developed to meet the following requirements:

- a) It must detect most EDAs including silent EDAs.
- b) It must be able to work directly on the sensors.
- c) It must prioritize energy efficiency, consuming a minimum of energy from the sensors.

To meet these requirements, the architecture we propose employs two modules: the Detection Module (DM) is responsible for monitoring and detecting the EDAs; the Learning Module (LM) is responsible for learning steps, where a learning algorithm analyses energy data and defines the key parameter that represents expected behaviours. Both DM and LM are deployed directly in sensors, making our proposal autonomous. In contrast to previous approaches (Alsirhani et al., 2022; Jan et al., 2019; Pu, 2019), our proposal monitors sensor energy consumption, applying distance metrics to identify anomalies in such consumption.

The rest of the paper is organized as follows. Section 2 presents a literature review of security on LoRaWAN security and intrusion detection of EDAs. In Section 3, we provide a detailed description of our proposed architecture. Section 4 presents the performance evaluation and results. Finally, Section 5 summarizes the conclusions and future work.

2 LITERATURE REVIEW

We provide a brief literature review of security in LoRaWAN (Subsection 2.1) and discuss the current intrusion detection of EDAs (Subsection 2.2).

2.1 Security of LoRaWAN

In an initial investigation, Nguyen et al. (2019) detailed an exhaustive examination of diverse attacks

on LPWANs. This study provided a comprehensive literature review, encapsulating research endeavours that explored the impacts of various attacks, including EDAs. The authors categorized these attacks based on the network layers, which encompass physical layer attacks like jamming, link layer attacks such as sleep cycle manipulation, and application layer attacks such as vulnerabilities in applications. The findings of this study illuminate the breadth and diversity of attacks associated with EDAs.

In a separate investigation, the researchers outlined LoRaWAN security features in detail (Yang et al., 2018). These features encompass channel confidentiality, the network join protocol, authenticity, and integrity validation. The study also scrutinized potential attacks on these features, including replay attacks, eavesdropping, and bit-flipping. Additionally, Mikhaylov et al., (2019) undertook an empirical validation of EDAs on a LoRaWAN device, providing insights into their potential ramifications. The experimental study revealed that EDAs aim to augment the transmission (TX) or reception (RX) of network data in sensors. This objective is pursued through two primary methods: firstly, a Denial of Service (DoS) attack, which results in channel overload, compelling sensors to transition to higher transmission power; secondly, the compromise of acknowledgment packets, compelling sensors to retransmit their packets. The researchers conducted empirical tests to substantiate the efficacy of these attack strategies.

In a correlated investigation, delineated by Neshenko et al. (2019), a thorough analysis of IoT vulnerabilities was undertaken. Certain vulnerabilities identified in this study can be leveraged for analogous purposes, notably in the context of "silent attacks" aimed at evading detection by maintaining a limited network footprint. These silent attacks include the compromised node scenario, wherein an attacker exploits vulnerabilities to initiate buffer overflows or attain privileged access. This enables the execution of malicious code, depleting sensor energy without generating network activity. Additionally, they encompass modified firmware attacks, where malevolent code is employed to compromise the sensor's lifespan. Moreover, the continuous advancement of IoT services by both industry and academia, exemplified by the development of Application Programming Interfaces (APIs) to provide diverse functionalities (Tzavaras et al., 2023), introduces new dimensions to IoT systems. While these advancements are beneficial for the industry, they concurrently introduce potential vulnerabilities, thereby rendering IoT systems susceptible to zero-day attacks, including EDAs.

2.2 Intrusion Detection of EDAs

In the existing literature, predominant attention has been directed towards the detection of EDAs in LLNs. For instance, Alsirhani et al. (2022) introduced the DISAM scheme, specifically tailored to mitigate the Span DIS attack, a threat that depletes the energy reserves of legitimate nodes in LLNs. Similarly, Pu (2019), the authors devised a scheme targeting a vulnerability in the Routing Protocol for LLNs (RPL). Their approach involved monitoring the packet reception count at a sensor. Additionally, (Jan et al., 2019) presented a lightweight IDS incorporating supervised machine learning, notably a Support Vector Machine (SVM), to identify adversaries attempting to introduce unnecessary data into the network, thereby detecting potential DoS attacks. It is noteworthy that these solutions primarily centre on the analysis of network traffic.

Concurrently, several contributions have explored the utilization of energy consumption analysis for the detection of attacks. For instance, Lee et al. (2014) introduced a lightweight intrusion detection scheme, employing energy consumption analysis to identify DoS attacks in networks employing 6LoWPAN. Han et al. (2013) proposed an intrusion detection scheme, relying on predictions of sensor energy consumption to discern various attack types, including flooding attacks and assaults on routing protocols in cluster-based Wireless Sensor Networks (WSN). Additionally, Proto & Carvalho (2020) delved into the application of three statistical distance metrics (Sibson, Hellinger, and Euclidean) to detect anomalies in the sensor energy consumption of a WSN. They presented a detection algorithm implemented in the sensors and conducted simulations, elucidating outcomes pertinent to the identification of EDAs triggered by flooding. Collectively, these endeavours underscore the significance of energy consumption analysis as a viable approach for detecting diverse attacks, including those relevant to LoRaWAN and Internet of Things (IoT) environments.

3 LIGHTWEIGHT ARCHITECTURE FOR DETECTING EDAS (LADE)

We propose an approach which entails deploying the Learning and Detection Modules (LM and DM) directly within the sensors. While it is conceivable to situate the LM in a network server to alleviate sensor

processing, we emphasize that the network communication between the two modules could potentially consume more energy than maintaining them within the sensors. This is because the energy expenditure associated with the transmission (TX) and reception (RX) operations in a sensor surpasses that incurred solely during processing, as presented in Section 4. Furthermore, we introduce a straightforward and resource-efficient learning algorithm, designed to minimize the demand on processing resources. The LADE and the integration between LM and DM are depicted in Figure 1.

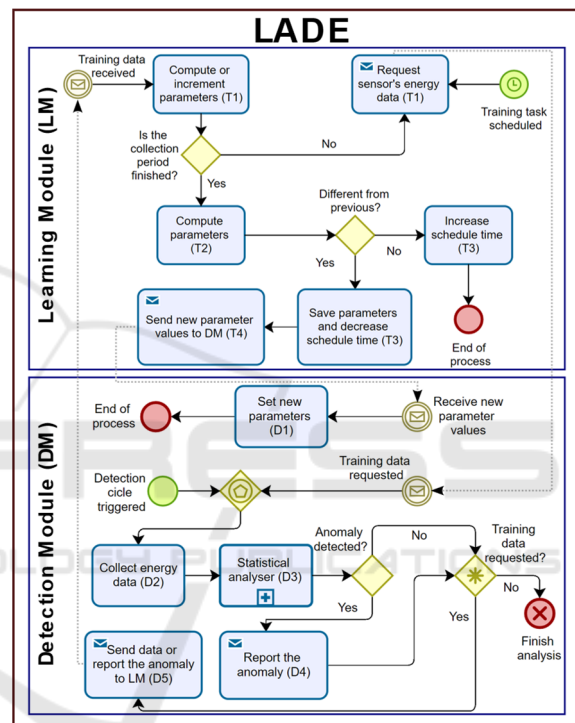


Figure 1: The LADE scheme and its modules.

Each module implements a different phase of the system as described in subsection 3.1. We also discuss the proposed algorithms in subsection 3.2.

3.1 Phases Description

LADE comprises two defined phases: the learning phase (Subsection 3.1.1) and the detection phase (Subsection 3.1.2).

3.1.1 Learning Phase

During this phase, the LM computes a key parameter used in the detection algorithm of the DM. Upon system initiation, it performs the first learning cycle to establish initial parameters. Subsequently, the LM

defines a random interval for the periodic execution of this task, which dynamically adjusts over time. This strategy aims to prevent attackers from discerning the training schedules and attempting to manipulate the energy consumption pattern. We define the steps of the learning phase as follows:

- a) *Request energy data*: The LM requests energy data from DM for a random period (T1).
- b) *Update parameters*: The LM computes the parameters (T2) using statistical analysis and, if the results are different from previous ones, then it sends the new parameters to DM (T4 and D1).
- c) *Update schedule time*: The algorithm updates the scheduled time of the learning phase based on the frequency of parameter changes. The less frequently the learning algorithm changes the parameters, the longer the scheduled time, and vice versa. Thus, it increments or decrements the scheduled time by a random percentage of the current value (T3).

Note that the learning phase is measured in minutes, as LoRaWAN applications typically operate with extended intervals between transmissions to conserve energy. Besides, it is advisable that the device stores parameter information in flash memory whenever possible, aiming to mitigate prolonged pauses in detection activity during restart operations.

3.1.2 Detection Phase

In this phase the DM is responsible for collecting energy data and applying the distance metric to detect anomalies. The steps of the detection phase are described as follows:

- a) *Set a key parameter*: Receive from LM and set a key parameter of the detection algorithm (D1).
- b) *Run the statistical analyser*: The DM analyses energy consumption data (D3) on-the-fly.
- c) *Report the LM of an intrusion detection*: If an anomaly is detected, then it forwards a report and the energy consumption data to the administrator or some system logger. Else, it takes no action (D4 and D5).

The detection phase only sends messages through the network in case an intrusion is detected. Currently, our proposal does not focus on mitigation attacks, which should be addressed in future works.

3.2 Algorithms Description

For clarity, we describe the detection algorithm in Subsection 3.2.1 and the learning algorithm in Subsection 3.2.2.

3.2.1 Lightweight Detection Algorithm

We adopted the methodology proposed by Proto & Carvalho (2020) as a baseline to meet the requirements outlined in the Section 1, with some enhancements. The proposal is solely an exploration of the application of distance metrics in energy consumption data. Its algorithm is limited by fixed parameters, making it less scalable and confined to WSNs. Such limitations have been addressed through the incorporation of a learning phase, beyond improvements on performance and autonomy.

A statistical distance metric quantifies the distance between two probability distributions. The algorithm proposed applies a distance metric called Sibson (Proto & Carvalho, 2020), which is based on Kullback-Leibler divergence and is defined in (1). Kullback-Leibler is not symmetric, which means $D(p,q) \neq D(q,p)$. Thus, Sibson combines such divergence to resolve the asymmetry (2).

$$D(p, q) = \sum_{x=1}^{\eta} p(x) \log_2 \frac{p(x)}{q(x)} \quad (1)$$

$$D_s(p, q) = \frac{1}{2} \left\{ D \left[p, \frac{1}{2}(p+q) \right] + D \left[q, \frac{1}{2}(p+q) \right] \right\} \quad (2)$$

We propose the Algorithm 1 to address the detection phase. The parameter δ is the threshold that defines an anomaly, while parameter λ denotes the expected value of sensor energy consumption for every two seconds, playing a crucial role in probability distribution calculations. In this study, we opt for the Poisson distribution, as suggested in Proto & Carvalho (2020). Nevertheless, we transform the energy data k by aggregating samples for each ω Joules (J), as defined in (3). This data transformation is essential for converting decimal energy data into integers and maintaining a restricted range of samples, thereby ensuring efficient distance calculations, as empirically observed. Other variables are defined as follows: N is the window size of an energy sample; A , B , P_A , and P_B are arrays designed to store energy consumption data and their respective Poisson distribution.

$$f(k) = \frac{1}{\omega} k + 1 = x, \quad x \in \mathbb{N} \quad (3)$$

We describe the Algorithm 1 steps as follows:

- *Collecting step*: the algorithm collects samples of energy consumption. When the collecting phase is activated, it collects two sets of samples with size N before going to the next step. The system collects an energy sample for every two seconds. When $N-1$ consecutive samples are

less than λ , the task is interrupted, saving processing and energy of the sensor.

- *Pre-processing step*: the algorithm converts the sets of samples into sets of probability distributions.
- *Intrusion detection step*: the algorithm applies the Sibson metric to calculate the distance between P_A and P_B . If it is less than δ , the anomaly is reported.

Data: energy consumption samples

Result: to detect and report an anomaly;

```

while true do
    read energy sample k;
    convert k to x using f(k);
    if x > λ or analyse is true then
        while slots A or B is not full do
            save x in slot A or B;
            if LM did not request data and
                the last N-1 samples x < λ then
                    break and go back to the beginning;
            end
            read energy sample k;
            convert k to x using f(k);
        end
        calculate PA and PB;
        calculate Sibson distance D(PA, PB);
        if D(PA, PB) < δ then
            report the anomaly;
        end
        if LM requested data then
            send data or report the anomaly to LM;
        end
    end
end
    
```

Algorithm 1: Lightweight detection algorithm of DM.

3.2.2 Learning Algorithm

Initially, LM accounts for computing the best value for the key parameter λ previously described. Thus, we propose Algorithm 2 which implements the LM scheme presented in Figure 1. The other variables used by algorithm are described as follows: $T_{current}$ and $T_{schedule}$ are respectively the current time and the scheduled time for the learning task; $T_{collect_period}$ is the period in minutes that energy samples must be collected; $\hat{\lambda}_n$ and σ_n are respectively the median and standard deviation of sample set with size $2N$; $\check{\lambda}$ and $\check{\sigma}$ are respectively the expected value and standard deviation calculated in the last cycle; $\bar{\lambda}$ is the mean of set E with all calculated $\check{\lambda}_n$ and; $\bar{\sigma}$ is the mean of set F with all calculated σ_n .

We describe the Algorithms 2 steps as follows:

- *Collecting step*: when the current time $T_{current}$ reaches the scheduled time $T_{schedule}$, LM requests to DM a set of size $2N$ of energy samples. For each set, the algorithm calculates the median $\hat{\lambda}_n$ and standard deviation σ_n . This step is repeated for $T_{collect_period}$ minutes.
- *Calculation step*: the algorithm calculates $\bar{\lambda}$ and $\bar{\sigma}$. Thus, it changes the value of λ only if $\bar{\lambda}$ is greater or less than $\check{\lambda} \mp \check{\sigma}$ calculated previously in the last cycle. Furthermore, λ has its value incremented or decremented by one unit. This technique serves to prevent unexpected behaviours or attackers from manipulating the learning phase by attempting to abruptly increase the λ value, thereby avoiding any consequential manipulation of intrusion detection outcomes.
- *Final step*: If λ changed after calculation, then send the new value to DM. After that, it calculates the new $T_{schedule}$ value randomly as described in item c) of Subsection 3.1.1.

Data: energy consumption samples

Result: new value of λ ;

```

while true do
    if Tcurrent = Tschedule then
        while Tcollect_period is not finished do
            request sensor energy data;
            calculate median value λ̂n;
            calculate standard deviation value σn;
            save λ̂n in array E and σn in array F;
        end
        calculate λ̄ as the mean of E;
        calculate σ̄ as of mean of F;
        if λ is not defined then
            do λ̄ = λ̄, λ = f(λ̄) and σ̄ = σ̄;
        else if λ̄ > (λ̄ + σ̄) then
            do λ = λ + 1, λ̄ = λ̄ + σ̄ and σ̄ = σ̄;
        else if λ̄ < (λ̄ - σ̄) then
            do λ = λ - 1, λ̄ = λ̄ - σ̄ and σ̄ = σ̄;
        end
        if λ is changed then
            send new parameters to DM;
            decrease Tschedule by a random value;
        else
            increase Tschedule by a random value;
        end
    end
end
    
```

Algorithm 2: Learning algorithm of LM.

4 SIMULATION AND RESULTS

In our simulation we adopted the energy consumption model presented in equation (4). The model defines four states of energy consumption in a sensor: CPU state, when it is processing data; SLEEP state, when it is in low power mode; TX state, when it is sending some data over the network; and RX state, when it is receiving some data through the network.

$$E_{all} = E_{cpu} + E_{sleep} + E_{tx} + E_{rx} \quad (4)$$

Few simulators fully support LoRaWAN, as it is a relatively recent technology up to this point. Thus, initially we implemented the LADE in C language and used FloRa (Slabicki et al., 2018), a framework implemented for OMNeT++ simulator, to collect the variables E_{tx} and E_{rx} of sensors. However, FloRa does not supply information about E_{cpu} and E_{sleep} , thus we calculated them based on microcontroller datasheets used in LoRaWAN. We describe the experimental setup, results, and discussions in Subsections 4.1, 4.2, and 4.3, respectively.

4.1 Experimental Setup

Our simulation deployed eight LoRaWAN nodes class A, repeatedly transmitting over 500 bytes of data at random sleep intervals ranging from 8 to 30 seconds. These nodes transmitted data to a LoRaWAN gateway connected to the network server via an IP network. The values for Spreading Factor (SF), Transmission Power (TP), and bandwidth (BW) were set to 12, 14dBm, and 125kHz, respectively. The LoRa protocol was configured to await acknowledgment (ACK) packages from the network server after a transmission and to retransmit data up to 15 times. The sensors' battery supplied 3.3V and we referenced the MSP430FR5969 microcontroller datasheet (Instruments, 2018) for E_{cpu} and E_{sleep} calculations. The expected energy consumption for the sensors' states is detailed in Table 1.

In addition, we set the following variables of the detection algorithm as fixed values: $N=8$, $\omega = 0.25$, and $\delta = 0.2$, drawing from experiments conducted by (Proto & Carvalho, 2020) and empirical considerations. Furthermore, we set $T_{schedule} = 15$ and $T_{collect_period} = 5$ (minutes) for the learning algorithm.

Besides that, we proposed to simulate two distinct types of attacks. The first one involves a jamming attack, wherein an attacker generates noise to compel sensors to retransmit data. The second type is a compromised node attack, wherein an attacker deploys malicious code to induce a continuous

processing state and to manipulate the LoRa protocol to request the RX state whenever the sensor is idle. In summary, we simulated two scenarios:

- 1) *Scenario without attacks*: We conducted a 30-minute simulation without any attacks, allocating 5 minutes for the initial learning phase and dedicating the remaining 25 minutes to the evaluation of intrusion detection. This scenario aims to assess the false positive rate.
- 2) *Scenario with attacks*: We conducted a 25-minute simulation for both proposed attacks to evaluate detection accuracy. The simulation used the same random transmission time employed in Scenario 1 for consistency, facilitating meaningful comparisons between scenarios. Consequently, the DM could complete the detection cycle up to 48 times, considering the window time N .

Table 1: Values of energy consumption in Joules/sec.

Node state	Expected consumption
CPU state	0.0054384
SLEEP state	0.00000231
TX state	0.14519995
RX state	0.03201

4.2 Results and Discussions

All results presented in this subsection stem from the mean of the output data from the eight sensors. Despite this aggregation, the simulation of multiple sensors is crucial to enable the assessment of varying frequencies of energy status changes, considering sensors transmitting data at distinct intervals.

In the first learning cycle of scenario 1, the LM returned a value of $\lambda = 2$, and this value was subsequently utilized as the starting point in scenario 2. Due to the experimental setup, the LM executed another learning phase only once. In the event of an anomaly detection, the LM discarded the energy samples provided by the DM for learning. Consequently, most sensors did not complete all $T_{collect_period}$ instances during the 25-minute simulation. Nevertheless, in Figure 2, we present the mean, median, and standard deviation of simulations with and without attacks to assess potential variations in λ over time. Despite an increase in energy consumption in Scenario 2, the median calculated did not exceed the criteria established by our methodology. Hence, the value of λ would remain unchanged throughout the simulation, even if it continues for a longer duration.

We evaluate intrusion detection outcomes in the DM using the F-measure. Employing the proposed

simulations, we compute precision, recall, and the F1 score, as presented in Table 2. Thus, we define a False-Positive (FP) event when the DM completes the detection cycle and reports an anomaly in a scenario without an attack. Similarly, a False-Negative detection (FN) occurs if the DM fails to complete a detection cycle or do not report an anomaly following a detection cycle, despite the simulated attack. Conversely, True-Positive (TP) and True-Negative (TN) events denote the opposite scenarios, respectively. Table 3 presents the mean values for each event in the simulated attacks. The obtained results revealed a high detection rate with minimal false positives, underscoring the potential effectiveness of our proposal against EDAs.

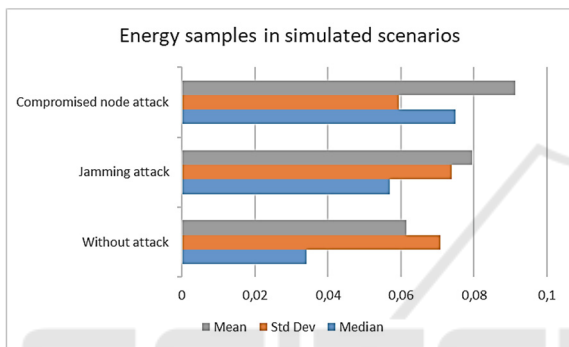


Figure 2: Statistical data of energy samples (Joules).

Table 2: F1-score of simulated attacks.

Type of attack	Precision	Recall	F1-score
Jamming attack	0.983	0.884	0.930
Compromised node attack	0.982	0.868	0.921

Table 3: Mean of events in the simulated attacks.

Type of attack	FN	TP	FP	TN
Jamming attack	5	38	1	45
Compromised node attack	13	33	1	45

Furthermore, Figure 3 illustrates the battery lifetime prediction in scenarios without and with LADE within sensors. To assess energy efficiency, we depict the battery lifetime of a sensor with LADE in two scenarios: first, we force the execution of all detection cycles, but without sending any reports; second, our system executes all detection cycles and sends reports in all cycles. Consequently, even in challenging scenarios, the energy consumption of LADE in the first scenario is negligible. In the second scenario, our system consumes only 0.3% more energy per cycle compared to other simulations, attributed to the transmission required to report the anomaly. This level of energy efficiency is

particularly notable, indicating the effective performance of the system.

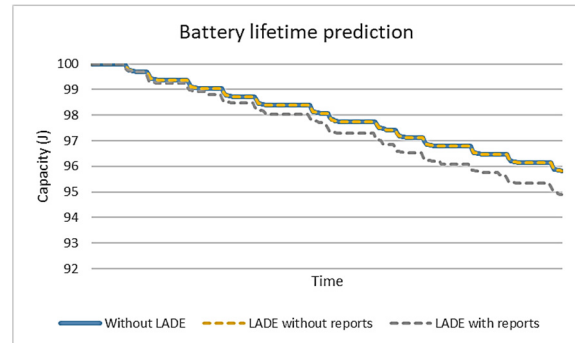


Figure 3: Performance comparison of energy efficiency.

Finally, we provide comparisons with LADE and other contemporary studies in Table 4, concentrating on diverse aspects relevant to the requirements delineated in Section 1. It is noteworthy that some of the works have the potential to detect multiple types of EDAs, albeit with a predominant focus on network-based attacks such as flooding or jamming. However, most of these works exhibit limitations in detecting silent EDAs. In addition, such proposals furnish information regarding energy efficiency.

Table 4: Comparison with recent works. Label: Y – Yes, N – No, P – Possible, NA – Not Available.

Requirements	Recent works					
	LADE	(Alsirhani et al., 2022)	(Pu, 2019)	(Jan et al., 2019)	(Lee et al., 2014)	(Han et al., 2013)
Detect more than one EDA	Y	N	P	Y	P	Y
Detect silent EDAs	Y	N	N	N	NA	NA
Deployed at sensors	Y	Y	Y	Y	Y	Y
Energy efficient	Y	NA	NA	NA	NA	NA

5 CONCLUSIONS

We have introduced a lightweight architecture designed for the detection of energy depletion attacks (EDAs) in LoRaWAN networks, denoted as LADE. Our architecture leverages distance metrics to identify anomalies in energy consumption samples within a sensor, incorporating two modules deployed on the sensor. In addition, the system employs an autonomous statistical learning algorithm to determine the optimal parameter for the intrusion

detection task. In our experimental setup, we achieved promising initial results, showcasing the high accuracy and energy efficiency of the proposed system. Furthermore, a comparative analysis with current research reveals our innovative approach to detecting both common and silent EDAs. The latter refers to situations in which an attacker compromises sensors through vulnerabilities, depleting sensor energy without generating network traffic.

Despite the obtained results, this work is currently in progress and requires further refinement. Primarily, we aim to enhance the learning phase by incorporating the configuration of additional parameters such as N , ω , and δ . This modification is intended to render the system more adaptive to different scenarios. Secondly, there is a need to improve the report-sending task of the detection module to prevent excessive communication in cases of consecutive anomalies. It is crucial to address the potential misuse of our current solution by an attacker to generate additional traffic, leading to the unnecessary energy waste of sensors. Thus, a solution must be devised to mitigate this risk. Lastly, we intend to propose an autonomous mitigation technique deployed at sensors that is not dependent on communication with external devices.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Thanks also to FAPESP MCTIC/CGI (Research project 2018/23097-3).

REFERENCES

- Alsirhani, A., Khan*, M. A., Alomari, A., Maryam, S., Younas, A., Iqbal, M., Siqqidi, M. H., & Ali, A. (2022). Securing Low-Power Blockchain-Enabled IoT Devices Against Energy Depletion Attack. *ACM Transactions on Internet Technology*. <https://doi.org/10.1145/3511903>
- Han, G., Jiang, J., Shen, W., Shu, L., & Rodrigues, J. (2013). IDSEP: A novel intrusion detection scheme based on energy prediction in cluster-based wireless sensor networks. *IET Information Security*, 7(2), 97–105. <https://doi.org/10.1049/iet-ifs.2012.0052>
- Instruments, T. (2018). *MSP430FR596x*, *MSP430FR594x Mixed-Signal Microcontrollers*. <https://www.ti.com/lit/ds/symlink/msp430fr5994.pdf>
- Jan, S. U., Ahmed, S., Shakhov, V., & Koo, I. (2019). Toward a Lightweight Intrusion Detection System for the Internet of Things. *IEEE Access*, 7, 42450–42471. <https://doi.org/10.1109/ACCESS.2019.2907965>
- Kuaban, G. S., Gelenbe, E., Czachórski, T., Czekalski, P., & Tangka, J. K. (2023). Modelling of the Energy Depletion Process and Battery Depletion Attacks for Battery-Powered Internet of Things (IoT) Devices. *Sensors*, 23(13). <https://doi.org/10.3390/s23136183>
- Lee, T.-H., Wen, C.-H., Chang, L.-H., Chiang, H.-S., & Hsieh, M.-C. (2014). *A Lightweight Intrusion Detection Scheme Based on Energy Consumption Analysis in 6LoWPAN*. Springer, Dordrecht. https://doi.org/10.1007/978-94-007-7262-5_137
- LoRa Alliance. (2020). *LoRaWAN 1.0.4 Specification Package*. https://lora-alliance.org/resource_hub/lorawan-104-specification-package/
- Mikhaylov, K., Fujdiak, R., Pouttu, A., Miroslav, V., Malina, L., & Mlynek, P. (2019). Energy attack in Lorawan: Experimental validation. *ACM Proceedings of the 14th ICARS, November 2020*. <https://doi.org/10.1145/3339252.3340525>
- Neshenko, N., Bou-harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-scale IoT Exploitations. *IEEE Communications Surveys & Tutorials*. <https://doi.org/10.1109/COMST.2019.2910750>
- Nguyen, V. L., Lin, P. C., & Hwang, R. H. (2019). Energy depletion attacks in low power wireless networks. *IEEE Access*, 7, 51915–51932. <https://doi.org/10.1109/ACCESS.2019.2911424>
- Proto, A., & Carvalho, T. C. M. de B. (2020). Applying distance metrics for anomaly detection of energy-based attacks in IoT sensors. *Brazilian Journal of Development*, 6(11), 92412–92435. <https://doi.org/10.34117/bjdv6n11-595>
- Pu, C. (2019). Energy Depletion Attack Against Routing Protocol in the Internet of Things. *16th IEEE CCNC 2019*, <https://doi.org/10.1109/CCNC.2019.8651771>
- Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys & Tutorials*, 19(2), 855–873. <https://doi.org/10.1109/COMST.2017.2652320>
- Sciancalepore, S., Tedeschi, P., Riasat, U., & Pietro, R. DI. (2021). Mitigating Energy Depletion Attacks in IoT via Random Time-Slotted Channel Access. *2021 IEEE CNS 2021*, 10–18. <https://doi.org/10.1109/CNS53000.2021.9705038>
- Slabicki, M., Premsankar, G., & Di Francesco, M. (2018). Adaptive configuration of lora networks for dense IoT deployments. *IEEE/IFIP NOMS 2018: Cognitive Management in a Cyber World*, 1–9. <https://doi.org/10.1109/NOMS.2018.8406255>
- Tzavaras, A., Mainas, N., & Petrakis, E. G. M. (2023). Internet of Things OpenAPI framework for the Web of Things. *Internet of Things*, 21(August 2022), 100675. <https://doi.org/10.1016/j.iot.2022.100675>
- Yang, X., Karampatzakis, E., Doerr, C., & Kuipers, F. (2018). Security vulnerabilities in LoRaWAN. *Proceedings - ACM/IEEE IoTDI 2018*, 129–140. <https://doi.org/10.1109/IoTDI.2018.00022>