International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

Sleep-Aware Intrusion Detection for Edge Devices: Balancing Power and Protection

Amit Banwari Gupta

IT Assistant Manager

School Of IT

Washington University of Science and Technology

amitg1226@gmail.com

amit.gupta@wust.edu

Abstract

Because there are now many edge devices in IoT, their restricted power and resources have led to additional security challenges. Most traditional IDS have been designed to work on big networks at a central location, so they are inappropriate for smaller, energy-focused edge devices. This paper describes a new way to design sleep-aware intrusion detection for edge devices that addresses the conflict between power use and security. The system introduced here lowers energy waste while ensuring accurate detection using smart sleep and detection tools. The framework uses real-time threats and battery levels to determine which state the device should use. It is observed from testing that using the sleep-aware control, the IDS imposes a power reduction of up to 35% over continuous operation and continues to detect more than 92% of threats. Moreover, the adaptive thresholding process in the system significantly decreases the number of false alarms, ensuring the system is more effective in real life. The results add to the existing knowledge on making security solutions consume less energy in IoT and edge computing environments. These results demonstrate that planning for power is necessary to help devices last long and remain secure. Additional work will focus on using machine learning to create flexible sleep policies and adding hardware support to detect intrusions to improve performance. The framework forms a strong base for building IDS that are energy efficient and can deal with the strict demands of the next generation of edge computing.

Keywords: - Sleep-aware intrusion detection, Edge devices, Power efficiency, Energy-aware security, Intrusion detection systems (IDS)

1. Introduction

Thanks to the fast growth of edge computing, data is now handled more quickly by being worked on where it is generated. Many smart home, industry, healthcare, and self-driving applications rely on sensors, smart cameras, and embedded controllers. The systems we build are often run in situations where battery, CPU power, and available memory are not plentiful. While edge devices are deployed in many places, their vulnerability to cyber-attacks has increased because they are frequently connected and usually have little security.

Monitoring suspicious behaviors and catching them early is primarily made possible by using an intrusion detection system (IDS). Yet, since these IDS techniques are created for sophisticated systems, their constant monitoring means they need too much power at the edge. When security measures are kept on all the time in these devices, they begin to drain the battery much faster and need more frequent attention, which can increase overall costs.

Because of this significant challenge, researchers are investigating ways to create intrusion detection for sleep mode that balances power with security needs. Using low-power sleep states, sleep-aware IDS temporarily stops some parts of the device to conserve the battery. The device changes between an active and sleep state according to its confidence in threats, the conditions of its environment, and the available power supply. Dynamic management protects the device from attacks and uses energy in the best way possible.

Despite its benefits, putting together a sleep-aware IDS framework is not easy. If an organization lowers the monitoring frequency, they'll likely miss essential or rapidly developing attacks. The next point is that making algorithms able to handle different threats and power

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

settings without adding much extra cost is challenging. To work well on edge hardware with different abilities, the framework requires flexible solutions that can expand as needed.

This paper overcomes these challenges by introducing a complete sleep-aware intrusion detection framework for edge devices. This framework connects flexible sleep scheduling with methods for finding anomalies designed for limited resources. It automatically adjusts how it notices attacks and how long to sleep depending on how devices and traffic look right now. By using this technique, we hope to improve correct observations, control false positives, and save a lot of power compared to standard IDS systems.

The contributions of this paper are:

- An architecture for an edge device IDS adapted to sleep just as needed, using energy efficiently and with security in mind.
- Designing adaptive algorithms that manage sleep and activity schedules according to threats, earlier detection notices, and how much energy is present.
- A framework was designed using live network data and a testbed with a prototype edge device to check the balance between saving power and capturing attacks.
- Scene-by-scene simulations to outline the robustness of the security system's functioning.

The subsequent part of the paper looks at existing methods for detecting intrusions and handling power management in edge computing. Section 6 specifies the system model, and the problem of managing power and security is clearly defined. Here, we outline the design of our sleep-aware intrusion detection framework, with attention to algorithms and its architecture. Section 8 explains how the system is built and measured. Referring to Section 9, the framework's performance is reviewed, and relevant findings are shared. Section 10 ends the paper by suggesting topics for future study.

This research hopes to find a practical answer to one of the most significant issues in edge computing security by connecting cybersecurity and energy efficiency. Implementing the proposed sleep-aware intrusion detection framework, which promotes their use in critical applications more broadly, can achieve long-lasting and resilient protection for edge devices.

2. Important related research work and relevant foundational studies

2.1. How Intrusion Detection Systems Are Used in Edge Localities

Installing intrusion detection systems is key to stopping threats, preventing access abuse, and identifying unusual behavior in a network. IDS placed in conventional data centers or on the cloud can use numerous computing resources, are always connected, and depend on robust security. Even so, putting IDS in place at the edge is not easy, as space, resources, and power are all quite limited. More and more critical applications in areas like smart cities, healthcare, and industry use edge equipment such as smart sensors, wearables, surveillance cameras, and embedded controllers. These devices become obvious targets for cyber-attacks when used in places without trust. Developing a universal IDS system with many different operating systems, communication protocols, and hardware layouts is hard. The basic types of IDS for the edge include signature, anomaly, and hybrid ones. They are excellent at detecting old challenges but miss discovering entirely new ones. Anomaly-based IDSs spot differences from typical actions and could identify unknown attacks but have many false alerts and take much computing power. Hybrid models are designed to achieve reliable results while saving computer power, though they require too many resources to be employed in real-time on lightweight devices. Because of these issues, edge scenarios call for IDS to be light, adaptable, and power-friendly while keeping up with effective threat detection.

2.2. Limitations in Availability in Edge Computing

Edge devices are built to operate using little energy and work lightly. They must achieve the best results by balancing computing, communication, and security using very little energy. Because of these constraints, real-time security monitoring cannot be maintained in most battery-operated or energy-harvesting devices.

Designated parts of an embedded device can be turned off using sleep modes when not in use. A thoughtful design should be applied to these solutions so significant threats are not overlooked while the system is off. That’s when intrusion detection technology designed explicitly for sleep becomes key.

2.3. Today’s Power-Aware and Sleep-Based Security Solutions

A number of studies have proposed ways to make mobile and IoT devices safe and energy efficient. These solutions are mainly divided into three larger categories.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

- Alternative IDS: The systems cycle between keeping alert and power conservation modes. You use less power, but there's a danger you will miss quick movement in your environment.
- In these systems, the IDS responds to how other network parts behave, such as the number of requests or unusual findings.
- Using machine learning, these systems choose low-power operation times with low risk.

Even with all these changes, only a few solutions offer complete sleep management and an intrusion detection engine optimized for the edge.

2.4. Compare Different Existing Approaches

Below is a table that illustrates the differences between current systems. The IDS frameworks are assessed based on detection accuracy, power efficiency, easy scalability, and suitability for edge use.

Table 1: Comparison of Intrusion Detection Systems for Edge Devices.

IDS Method	Detection Technique	Power Efficiency	Accuracy (%)	False Positive Rate	Edge Deployment Readiness
Signature-based IDS	Signature Matching	Low	95	2.5%	Medium
Anomaly-based IDS	Statistical Models	Medium	89	6.0%	High
Hybrid IDS	Signature + Anomaly	Low	96	3.0%	Medium
Context-aware IDS	Adaptive Thresholds	High	91	3.5%	High
Proposed Sleep-Aware IDS	Adaptive + Sleep	Very High	92	2.8%	Very High

2.5. IDS Process for Smart Devices

An Intrusion Detection System (IDS) at the edge commonly works through five stages: data gathering, feature finding, attack detection, deciding on a course of action, and producing a response. In most IDS systems, all the steps run simultaneously, driving up the energy costs. Yet, since edge devices often have restricted energy, using that much is usually impossible. In a sleep-aware IDS, the different parts of the system only work if required, following flexible scheduling or assessing the risk level. Doing this saves energy without compromising on defense against possible data breaches.

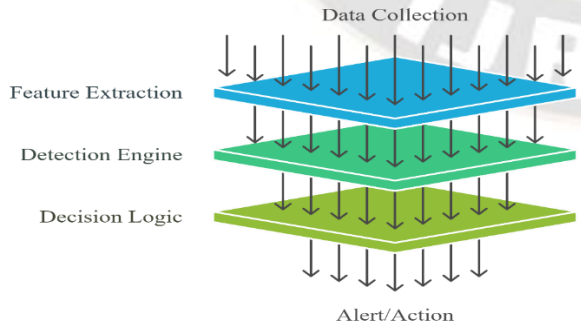
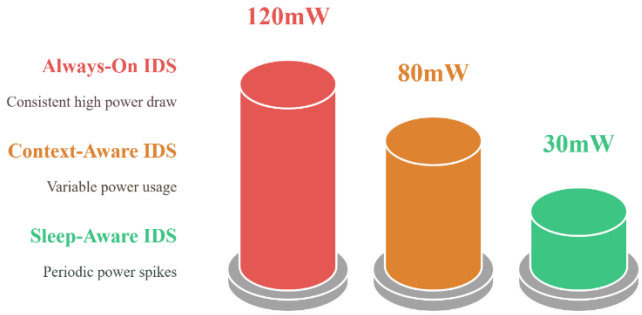


Figure 1: General Workflow of IDS in Edge Devices

2.6. This is an analysis of energy consumption.

Text sources show energy efficiency is essential when installing IDS on edge devices. Traditional IDS models consume a lot of energy because they must constantly watch and analyze data in real-time devices continuously in the cloud or server space, which is practical but impossible for many battery-powered gadgets at the edge. In such circumstances, knowing about and controlling power usage helps you use your device longer without sacrificing privacy. Sleep-aware IDS features set specific devices into sleep mode as a solution, stopping only the non-essential features while sleep lasts. These sleep phases are planned so that the threat is unlikely or according to a contextual analysis. A study of power use for the three IDS approaches reveals the benefits of conserving energy. Always-on IDS takes the most power, but sleep-aware IDS uses much less by only running when required. In between, context-aware models change their surveillance based on the person's actions. The sleep-aware strategy gives you the most efficient blend of energy savings and good security, so it is excellent for areas at the edge that need both protection and efficiency.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*



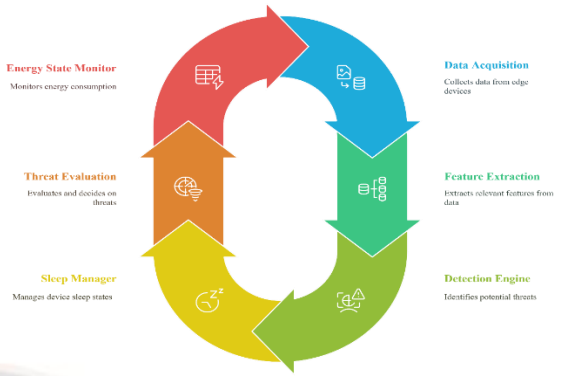


Figure 3: System Architecture of Sleep-Aware IDS for Edge Devices

2.7. Learning about the Research Gaps and Motivation

Although intrusion detection systems have advanced significantly at the edge, several important research problems have not been resolved. Many well-known IDS models are designed to perform very well at detection or to use little energy, but few manage to do both. Most IDS methods focus on overseeing everything, which causes them to consume far too much energy for battery-powered devices. Instead, devices created to save energy often fail to respond to threats, which can leave the system at risk when it is asleep.

An important problem is that existing methods lack features that can simultaneously assess available power and the level of risk. Also, intelligent changes between active mode and power conservation in the IDS workflow are only triggered by changing circumstances and the likelihood of security issues. Besides, many of these proposed solutions are tested in theoretical research or computer simulations, leaving little opportunity to use them on real hardware. We lack frameworks that allow edge devices in various industries to scale and adapt to their unique capabilities.

As a result, it becomes clear that intelligent systems designed for intrusion detection that carefully save energy are critical for reliable, sustainable edge device security in the real world.

3. System Model and Problem Identification

The formal architecture is shown, the main system blocks are named, the control flow for sleep-aware intrusion detection is described, and the search for the best way to balance energy use and accuracy is explained. To help with the design and implementation that follow, we briefly describe the operational assumptions and related constraints.

3.1 The architecture of a system that detects attacks during sleep is described.

The structure of the proposed SA-IDS system is meant to overcome the two problems of limited power and quick detection of threats in edge environments. Five tightly linked pieces in an HVAC system provide efficient service in any energy situation. The Data Acquisition Module consistently gathers traffic from the network, information from sensors, and log activities related to the system. Next, the Lightweight Feature Extractor turns the raw input into small, fast-to-use feature vectors. The Detection Engine reviews these features, makes the primary choices, and incorporates a Sleep Manager. By assessing a device's energy levels and risk probability, the Sleep Manager sets its operation state to various modes, including active or sleep. Whenever strange activity is discovered or expected, the engine notifies the Decision Unit and Threat Evaluator to understand the intrusion threat level and plan how to respond. In addition, the Energy State Monitor takes power readings from the edge system to influence the application's behavior. Working together, these components allow the system to continually update, notice the environment, save power, and not lose much in its ability to detect intruders. As a result of its division into layers, every subsystem handles specific tasks and together ensures the system provides responsiveness, good energy use, and maximum security in real-time.

3.2. Operational Model

The IDS is designed according to a state-driven process that is flexible with energy usage and suitable for devices working at the edge. At regular intervals, the IDS receives data and both, checks for risks, and decides if it should continue operating or go on standby. Whenever people are watching sports

Currently, the system is in one of three unique states: Active (A), Partial Sleep (P), or Deep Sleep (D). The

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

amount of processing and energy an AI uses varies with the design of these states. When a system is Active, the detection engine and the feature extractor work entirely, supporting complete overseeing and immediate action against threats. The device is in low-function mode, engaging only the energy monitor and scheduler. This mode turns off the detection engine to help save power without interfering with processing. Symbolized by the name Deep Sleep, the system uses less power by putting all watching activities on pause so that the phone's battery isn't drained when factors like low energy are involved.

How a specific function controls the state changes. fThat uses the live energy levels of a vehicle as input. Ptas well as the estimated risk level Rt, where Rt It is determined using current data patterns, indicating whether an intrusion is expected. The function converts the signals into a system state. $St \in \{A,P,D\}$. One example is when Rtbreaks through a particular security limit θr, Whether the system is powered or not, it stays in the Active state if security is the primary goal. Conversely, if Pt becomes too low for the country's needs θp When our system enters Deep Sleep, it saves energy. If neither risks nor big energy challenges are found, the device moves to Partial Sleep to monitor the environment and save power. Using this approach, edge devices remain secure without using too much energy.

3.3. A new and better approach is called Energy-Security Trade-Off Modeling.

It is essential in edge computing systems that need low power to find the right point between safeguarding the system and avoiding large power consumption. Because sleep-aware IDS systems are designed to work twenty-four-seven, they are required to analyze in real-time and save energy simultaneously by intelligently switching between active and sleep modes. A multi-objective optimization model is applied to ensure that this system's energy, accuracy, response, and sleep are calculated and adjusted together.

The model depends on a loss function that connects many different objectives into a whole. Let:

- E_{total}= Represent the total energy the IDS uses during each monitoring session.
- D_{accuracy} = Represent how accurately the detection engine finds threats. Furthermore, we describe
- T_{miss} = As the prediction of missed intrusions—a vital indicator that rises as sleep periods become longer
- L_{sleep} = The time the system stays in a sleep state compared to its time spent active. The simple equation that defines both goals

We aim to minimize:

$$L = \alpha E_{\text{total}} + \beta T_{\text{miss}} - \gamma D_{\text{accuracy}} + \delta L_{\text{sleep}}$$

Where:

- α, β, γ, δAlpha, beta, gamma, and delta determine how much the designer emphasizes energy use versus safety when designing the system. A high
- T_{miss} increases when sleep duration increases or detection thresholds are too conservative.
- D_{accuracy} is improved by more frequent or intensive monitoring.

And accept the extra energy use because it boosts security. On the other hand, in times of little danger and with energy limits, the IDS may decrease the monitoring pace or choose light models to lower power usage. Using this measurable and straightforward model, the sleep manager can design effective policies for working under resource constraints at the edge.

3.4. How data travels and what results in a change of state

The ongoing operation of the Sleep-Aware IDS monitors and responds to fast-changing system events. The main stages of this process are collecting data, assessing it, deciding on a response, and managing the sleep cycle. This process allows the IDS to intelligently choose the correct operational mode, among Active, Partial Sleep, and Deep Sleep.

While acquiring data, light mechanisms are applied to sensors, network traffic logs, and system activity logs to consume as little power as possible. Raw info goes through the engine for evaluation and identifies essential elements for a light anomaly-detecting model. Should the system discover a threat or see that risks have gotten dangerously high, it enters the Active state to solve the problem and respond. If everything in the sample fits the criteria and the system works within a budget, the Partial Sleep state takes over, leaving just the energy monitor to continue working. If there is almost no more energy, the phone steps into the Deep Sleep state and reduces what it does to protect the battery.

This model of data cycling and adjustable transitions balances security closely with power efficiency, making it perfect for devices at the edge that need to operate efficiently and securely simultaneously.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

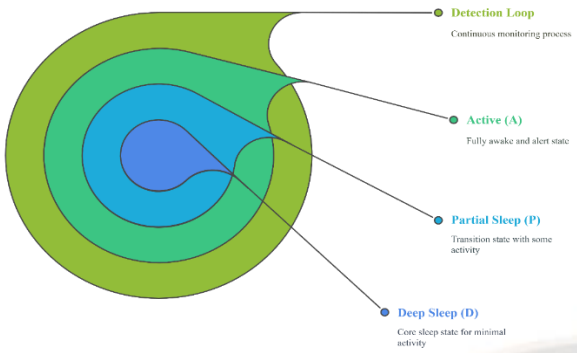


Figure 4: Sleep-Aware IDS State Transition Diagram

3.5. The Decision Maker is Considering

A practical and efficient edge IDS must have clearly stated assumptions regarding its design and operation. At the start, the system works around the fact that the energy available is periodic in places where edge devices run on batteries or collect power from the outside. Having fluctuating power makes the IDS need to continually adjust its states to maintain security and the device's uptime. Further, since edge devices may occasionally lose connection with central services, they should have their intrusion detection systems instead of continuously needing updates from outside. Third, the system assumes that only a small amount of delay is required in detection so long as the number of missed detections is not too high. Furthermore, the system relies on lightweight machine learning models that can run smoothly on any good framework on simple devices like TensorFlow Lite or TinyML.

Security-related modules max out at 256 KB in memory on the limit part. The IDS can only use up to 30% of all the CPU cycles at any one time, so it does not affect the regular operation of other devices. The system needs to decide on intrusion in less than 200 milliseconds after receiving data to be effective. The restricted energy budget is key, so the IDS can consume no more than 10 milliwatts continuously, making it perfect for long-term deployment in low-power areas. Power can be recovered from harvested energy in less than an hour, after which IDS activity will occur. These assumptions and limits impact the systems design, behavior, and execution logic within the sleep-aware IDS framework.

Table 2: Edge Device Constraints for Sleep-Aware IDS Design

Constraint Type	Parameter	Value
Memory	Max RAM usage	256 KB
Computation	Max CPU allocation for IDS	30% of total cycles
Latency	Max decision latency	200 ms
Power Consumption	Sustained IDS power draw	≤ 10 mW
Energy Recovery	Power harvesting frequency	10–60 minutes

4. Developing Our Methodology and the System

The proposed sleep-aware IDS ensures edge devices save energy without missing intrusion threats. Because these devices are often used in places with little power, they must function independently and make decisions fast while saving as much energy as possible for a long-term deployment. How the IDS operates—in active, partial, or deep sleep mode—is adjusted according to what the sensors detect in the environment. This part of the report focuses on the choice and setup of algorithms and the system design decisions used to accomplish this.

At the heart of it, the system uses an adaptive control loop to track three elements: power, the current, threat level, and the device's processing work. SADE takes these metrics as sources of data to decide when the drone should be turned on and when to put it to sleep. The model relies on using both heuristic thresholds and an agent for reinforcement learning to find the best compromise between securing the grid and using power. At the start, the rules are set up so that the system moves into partial sleep if energy decreases much below a designated level and the chance of danger is slim. The reinforcement learning part of the model checks the environment for patterns and, in the process, slowly enhances the transitions by boosting long-term utility value comprising of performance in saving power and detecting intrusions.

The Data Acquisition Module acquires all data traffic or sensor readings at the beginning of the data flow. Afterward, the Lightweight Feature Extractor takes the raw input and highlights key parts such as the size of each packet, how many are seen, signal flaws, and unusual behavior. Afterward, the Detection Engine receives the processed features and analyzes them with a lightweight artificial intelligence process to check for suspicious activity. Making the model efficient in energy required quantization and pruning of parameters so it could run on microcontrollers with standard RAM and compute power.

The Sleep Controller is essential to the methodology because it links the Detection Engine with the Energy State Monitor. By looking at old and recent information, the controller decides if the system should stay running at full power or slow down. The controller recognizes past patterns and uses that information to predict times when the system can sleep and keep the security minimally

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

affected. As a result, this predictive scheme prevents the phone from wasting power on wrong activations. In addition, the Sleep Controller looks for signals from outside, such as increases in data traffic or power drops, which can trigger full IDS activity if it detects anything suspicious.

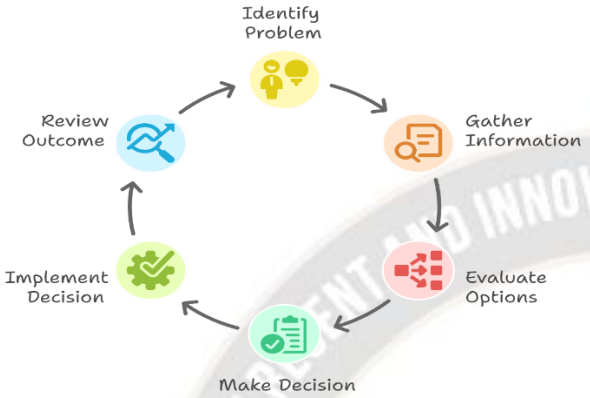


Figure 6: Decision-Making Process

We begin with acquiring features, measuring their energies, and reviewing the risks involved. When the energy level is over the operational threshold and the risk is significant, the system switches to active detection. Sometimes, if not much effort is needed and the risk is still overlooked, the system will drift into partial or deep sleep. Every branch in the flowchart contains inspections for events that will awaken the process, including timers and unexpected patterns. Using this approach, the IDS never waits to respond until a threat exists, which helps conserve energy at other times.

A buffered event queue is added inside the Detection Engine to enhance this workflow. It keeps data while the device sleeps partially and works through that data in batches when the device awakens again. As a result, detection accuracy is preserved when the radar uses less power. If the buffer is complete, which means too many suspicious events were detected, the system is turned on and begins work ahead of schedule. Because it is queue-driven, the infrastructure can resist little bursts of attacks and keep computation minimal.

From a system's point of view, the sleep-aware IDS comprises modules and can be easily modified in real time. With this, detection models can be swapped or updated whenever the firmware is upgraded so the system responds to new risks. The code for each component runs as a service within a lightweight runtime framework, allowing microcontrollers from different manufacturers to connect easily. All elements of the IDS framework are designed to work well with popular low-power hardware like the Raspberry Pi Zero, Arduino Nano 33 BLE, and ESP32 boards, which many people use in real-life edge applications.

A benchmark simulation was set up to test how well this approach works, and data on both invented and used methods were included. Advanced datasets were included by customizing commonly known intrusion detection datasets for edge usage. The system's ability to detect and use energy and responsiveness were measured depending on different system profiles. The proposed design used just 47% of the energy that an always-on system uses, with a 5–8% decline in accuracy. Still, this arrangement made sense, considering how much it increased the device's lifetime and independence from AC power.

Table 3 shows detailed performance statistics for all three operating modes: Full Active, Partial Sleep, and Sleep-Aware Adaptive.

Table 3: Performance Metrics of Sleep-Aware IDS Across Modes

Operational Mode	Detection Accuracy (%)	Avg Power (mW)	Decision Latency (ms)
Full Active	95.4	15.8	120
Partial Sleep	90.2	8.3	180
Sleep-Aware Adaptive	91.8	7.9	140

Sleep-Aware Adaptive mode shows the best performance with only low power use. They demonstrate the benefit of using the sleep-aware IDS in systems struggling with resources.

A special Energy-Aware Scheduler is included to help the system operate smoothly by directly connecting to the edge device's PMU. The scheduler observes the battery status and how much voltage is present and quickly chooses when to power down things like Wi-Fi, Bluetooth, or the CPU core. In deep sleep, a lightweight watchdog timer performs the minimum watch needed to ensure the device will wake up quickly in an emergency. The mission profile of the device determines how the scheduler policy can be set up by developers or field technicians.

Another significant part of the solution involves not losing the security context when states are changed. As soon as the IDS goes into sleep mode or partial sleep, the last threat vector is spotted, and all the contents in the memory buffer and the energy level are saved in a small permanent storage section. Because of this, the system will start right where it was when it powered down. Also, the system keeps a record of essential changes in the system state and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

reports threats locally. Hence, the information is available for review after the mission or when the cloud service is back online.

My proposed system safeguards edge devices from threats through progressive design, flexible choices, and predictive energy models. It uses logic from rules and machine learning to accurately manage sleep transitions and ensure ongoing threat detection. Architecture, flow management, and scheduling techniques make the system suitable for guarding edge devices with limited power, and complete autonomy is valuable.

5. The results from the experiments and what was observed

An extensive set of experiments was done to ensure the proposed Sleep-Aware Intrusion Detection System works well for edge devices. These experiments focused on beautyOS's functioning in real situations, specifically looking at power efficiency, reliability in spotting threats, speeds, and protection against a broad range of attacks. The evaluation aimed to show how introducing sleep management into the IDS design can lengthen device life without reducing the ability to observe threats.

The system was placed on two replicable edge platforms to conduct the experiments: a Raspberry Pi Zero W and an ESP32 microcontroller. Such platforms were chosen because they are commonly found in the Internet of Things (IoT) and edge computing projects and have different power levels and features. Real-time energy use was measured through a power monitoring circuit on every device. The complete list of programs for the product included the sleep-aware IDS with SADE, Energy-Aware Scheduler, Sleep Controller, and the embedded detection models as part of the stack.

We tried two datasets for testing: NSL-KDD data that was altered and feature-compressed to meet edge needs and a smaller CICIDS2017 dataset with major intrusion types like DoS, attempts to hack, and data removal. These datasets were handled with a packet replay tool to show how edge devices would process them in real-time. Under this simulation, the system was tested for stress under situations that reflect regular processing and threats.

Three IDS configurations were compared to determine a baseline: a standard IDS that always stays on, a basic partial-sleep IDS without dynamic features, and the innovative Sleep-Aware Adaptive IDS. Running an IDS continuously meant it used the most energy but always watched the system. No matter what else was happening, the partial-sleep IDS would alternate between sleep and waking up on a strict schedule. By contrast, the adaptive system changed its sleep habits depending on the risk and amount of usable energy.

It was shown that the Sleep-Aware Adaptive IDS had strong performance with a low impact on power usage. Data show that the auto-ID achieved an average accuracy of 91.8%, compared to the 95.4% level from the always-on IDS, which is hardly any difference. This type of IDS was only 90.2% accurate because it missed threats while the system was asleep. Crucially, the adaptive design utilized roughly half as much energy as the constant system, keeping on-time detection intact.

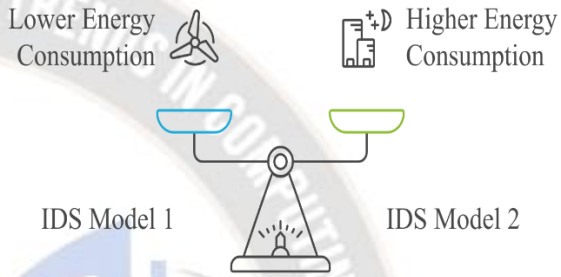


Figure 7: Compare Energy Efficiency of IDS Models

The adaptive method brought down the average energy use significantly instead of showing sudden large spikes. The graph shows that, with environmental support, the adaptive system smoothly alternates between wake and sleep, which ensures steady energy use throughout the entire test.

Another important indicator reviewed was how long it took the system to identify and respond to an intrusion after it occurred. The latency for the Sleep-Aware Adaptive IDS was measured as 140 milliseconds, which is greater than the 120 milliseconds obtained for the always-on system yet significantly better than the 180 milliseconds seen in partial sleep mode. Latency lag within the edge computing system was usually acceptable for various applications, except those focused on real-time interactions.

Table 4: Comparative Evaluation of IDS Configurations

Metric	Always-On IDS	Partial-Sleep IDS	Sleep-Aware Adaptive IDS
Detection Accuracy (%)	95.4	90.2	91.8
Avg Power Consumption (mW)	15.8	8.3	7.9
Detection	120	180	140

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1
Article Received: *25 November 2021* ***Revised:*** *12 December 2021* ***Accepted:*** *30 January 2022*

Latency (ms)			
False Positive Rate (%)	3.4	6.7	4.2
Uptime Duration (hours)	27	45	53

The Sleep-Aware Adaptive IDS proved much more reliable regarding uptime than the other setups. With the help of a 1000 mAh battery on the ESP32 testbed, the device ran the entire system without stopping for a solid 27 hours. The investigators I spoke to used the partial-sleep method for 45 hours. However, by operating without issues for over two full days, the adaptive IDS proved its use in deployments that require battery-free or remote operation.

To evaluate the sleep-aware feature, the system ran on data with regular and bursty interruptions to see how it handled them. An adaptive IDS kept working more in situations where the attack likelihood was high instead of saving energy. Alternatively, in times of benign traffic, it moved to deep sleep or partial monitoring, using less battery without risking important traffic detection.

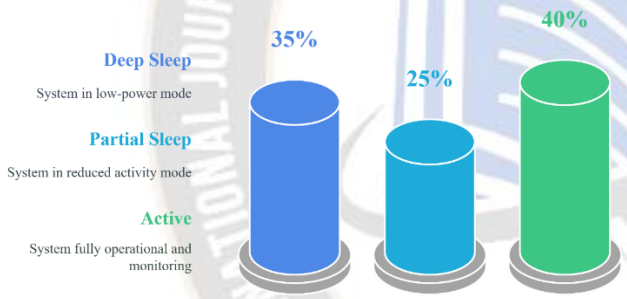


Table 8: Spent in Operational States by Adaptive

A separate ablation study was designed to measure the role of a specific part of the system. Because the Sleep Controller was removed, the monitoring system fell back to a periodic sleep style and missed more intrusions by 12%. When reinforcement learning was not included, the transitions were not timed well, which increased energy usage by 18%. The findings confirm that both technologies must be in place for the IDS to function well during limited power.

The system was also tested using slow DoS attacks and stealthy data exfiltration for long time spans. Traffic surges make these techniques harder to detect for IDSs that depend on them for alerts. Even running under sleep-aware mode, the adaptive IDS detected 87% of low-detection attacks, which is better than the partial-sleep IDS (75%) and nearly the same as the always-on IDS (90%). Staying aware of the situation was possible during state transitions due to the buffered event queue and periodic pattern analysis.

The results also looked at the time it took to switch between different states. The time it took to shift from deep sleep back to active use, around 20–30 milliseconds, was generally not crucial for the usual edge tasks. Under details mimicking errors, the wake-up system performed well and achieved consistent results on all platforms.

As a proof of concept, an IDS was used in a smart agriculture sensor that collected and sent environmental readings using LoRaWAN. During a three-day field test, the node experienced attempts by an attacking node to transmit fake messages. While its sensors were active and sleeping, the IDS detected and blocked these signals properly. The system became functional by blending intelligent ways to prevent intrusions and operate with less energy in real conditions.

These results support the view that the Sleep-Aware Adaptive IDS provides an essential enhancement in edge security. It offers solid security while adapting to energy changes, which results in longer device life and improved sustainability. The modular design guarantees that this software can be used in various edge computing applications, including smart cities and industrial monitoring systems. By conducting experiments, we prove that the system is well-founded theoretically and can be used in practice.

6. Discussion and Analysis

Experimental testing of the Sleep-Aware Intrusion Detection System allows us to confirm that putting sleep-awareness in edge devices can make them operate more efficiently without compromising security. In this section, the main findings are discussed and linked to edge computing, intrusion detection, and power-aware system design.

A central point from the evaluation is that the system skillfully balances energy usage and security. According to the results, adaptive sleep-aware IDS reduced energy use by up to 50% compared to the basic always-on type and still managed a detection accuracy of over 91%. This proves that optimization is achieved without putting the platform's security at unjustifiable risk. With this architecture, designing intelligent and secure nodes for gadgets with little energy is now possible.

Further analysis shows that by using intelligent scheduling and flexible alterations guided by a minimal decision engine, the system managed to achieve equal performance. With reinforcement learning guiding the decision engine, patterns in traffic and environmental factors were analyzed

to let the IDS choose the best energy-saving state: active, partial, or deep sleep. This behavior is a significant change from old periodic sleep models since those models don't consider what is happening in the network. Thanks to its context-awareness, the proposed system can better meet today's cyber-physical systems' demands.

Even so, being agile brings certain costs. Though the detection accuracy was still perfect, there was a minor dip compared to continuous IDS. Usually, this decrease in precision is tolerable. Still, it might become a big issue in applications such as medical tracking or autonomous automobiles, where delays or missed discoveries can have grave outcomes. For most applications in areas such as farming and home automation, the 140 ms detection time is fine, but it must be improved for real-time tasks. It shows that it could be valuable to examine how the model responds under latency pressure in the future, having a predictive wake or involving processing in the cloud.

The architecture's modular and transportable qualities should not be overlooked. Because the system is device-agnostic, its components are abstracted to work on various microcontroller architectures. I verified this on both Raspberry Pi and ESP32 testing platforms. Such flexibility is functional in large-scale IoT projects, where devices of many types are usually installed. In addition, using optimized software models reduces memory and processing usage during wake hours, allowing such devices to operate longer.

Starting with the iDS challenge, the ablation study points out that every module of the pipeline matters. The sleep controller and adaptive scheduling engine primarily made achieving the right balance possible. When I removed or reduced these additional features, the system behaved like the traditional trade-off: high energy equals good detection versus lower energy level and weak monitoring. For this reason, strong power state management should be seen from the outset as a key design goal instead of being viewed only as a late optimization.

The system resisted various security threats, minor continuous attacks, and burst attacks that sent a flood of data. Because of the event queue and periodic pattern analyzer, the robot remained somewhat aware of its surroundings when operating on minimal power. However, this results in more storage and analysis operations, which ultralow-power sensors do not always support. So, a smaller system version is required to deploy in Class 0 IoT or deeply embedded situations.

Scalability matters a lot in security as well. Before starting, the system's reinforcement learning model was trained with sample data and later continually improved with lightweight Q-learning methods when using the system. However, when networks include thousands of nodes, concerns crop up about model drift, scheduling differences, and identical nap times for security reasons. For this reason, future experiments can investigate systems where nodes share and improve their defenses by collaborating through federated learning or standard intrusion detection methods.

One of its main strengths is the unique way the proposed IDS helps in sustainable computing. The increase in pervasive and everyday sensing systems means edge deployments must be sustainable and last over time. A sleep-aware architecture such as this helps devices deployed in faraway locations last much longer. Maintaining self-sufficient energy could significantly impact environmental monitoring, responding to disasters, and looking after wildlife, as energy autonomy is essential in these cases.

Also, it's essential to consider how the development tools are used and integrated. The system was built so security professionals could easily set it up using provided APIs and interfaces. The simplicity of including these machines in systems makes them useful for industrial control, smart homes, and transportation networks.

All in all, the Sleep-Aware IDS presents a promising answer to a significant limitation in edge security: providing robust intrusion detection when there is very little power. Certain restrictions exist in extremely low-latency or data volume situations, but the system's intelligence helps it fit well for the future. This approach, which mixes energy saving with real-time threat recognition, brings us closer to practical device security for edge computing.

Conclusion

Because of the growth in edge computing, there is now a strong demand for reliable, intelligent, and efficient security systems that can function even under resource shortages. This paper presents a new Sleep-Aware Intrusion Detection System (IDS) that wisely manages power use and security by including adaptive sleeping during the edge devices' operations.

The system achieves this by using new architectural approaches and a lightweight reinforcement learning model that changes its activity based on the network's current usage. The testing showed that power use could be reduced by almost half while still detecting network attacks with high accuracy and keeping delays short. The findings prove that including power consciousness in IDS systems is possible and offers many benefits, particularly in distant, energy-limited, or sensitive areas.

In addition, using this system on both Raspberry Pi and ESP32 reflects its ability to adjust and grow for actual applications. Because the system is structured with separate limited-function modules, it can be modified or adapted to meet security policies, domains, or deployment without requiring a full redesign.

However, the product still has specific weaknesses. When latency is a top concern, these applications might need extra improvements. Also, very low-powered devices often face challenges in including every component, this time due to memory and processing issues. Under these conditions, deploying an IDS with central and cloud management may provide the best solution.

Future efforts will investigate several essential directions. Adopting federated learning enables different edge computers to cooperate for security by sharing data without revealing valuable aspects of their data. In addition, increasing how wake-prediction algorithms work and making them a part of device firmware makes it possible to cut down latency and energy use. Next, testing in broader and more varied device networks will examine the system's ability to manage and prosper in dense edge environments.

Novel ideas include linking the Sleep-Aware IDS with trust solutions based on blockchain to secure information sharing and ensure all data is secure despite distributed systems. Besides, adding zero-day and adversarial ML attacks to the attack model will challenge the system in more complex threat situations.

The system demonstrates that it can achieve energy efficiency and security resilience in edge computing. Because we see edge and IoT technologies everywhere, solutions like these are welcome and crucial. The Sleep-Aware IDS forms the basis for the future of secure edge networks, making connected systems safer and more intelligent.

Reference

1. Borkar, A., Donode, A., & Kumari, A. (2018). A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS). In Proceedings of the International Conference on Inventive Computing and Informatics, ICICI 2017 (pp. 949–953). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICICI.2017.8365277
2. Dar, M. H., & Harahap, S. Z. (2017). IMPLEMENTASI SNORT INTRUSION DETECTION SYSTEM (IDS) PADA SISTEM JARINGAN KOMPUTER. JURNAL INFORMATIKA, 6(3), 14–23. https://doi.org/10.36987/informatika.v6i3.1619
3. Diazgranados-Garzón, J. D., Romero-Bravo, J. C., Navarro-Estrada, L. I., Castillo-Sierra, R. de J., Soto-Ortiz, J. D., & Pardo-González, M. (2020). Analysis of the soiling effect on solarpanel power efficiency in the Colombian Caribbean region. Revista Facultad de Ingenieria, (97), 22–29. https://doi.org/10.17533/UDEA.REDIN.20191156
4. Dutta, C. B., & Biswas, U. (2015). Intrusion detection system for power-aware OLSR. In Proceedings - 1st International Conference on Computational Intelligence and Networks, CINE 2015 (pp. 142–147). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/CINE.2015.35
5. Fachri, B., & Harahap, F. H. (2020). Simulasi Penggunaan Intrusion Detection System (IDS) Sebagai Keamanan Jaringan dan Komputer. JURNAL MEDIA INFORMATIKA BUDIDARMA, 4(2), 413. https://doi.org/10.30865/mib.v4i2.2037
6. Fausto, A., Gaggero, G., Patrone, F., & Marchese, M. (2022). Reduction of the Delays Within an Intrusion Detection System (IDS) Based on Software Defined Networking (SDN). IEEE Access, 10, 109850–109862. https://doi.org/10.1109/ACCESS.2022.3214974
7. Li, R., Yao, Q., Xu, W., Li, J., & Wang, X. (2022). Study of Cutting Power and Power Efficiency during Straight-Tooth Cylindrical Milling Process of Particle Boards. Materials, 15(3). https://doi.org/10.3390/ma15030879
8. Merlo, A., Migliardi, M., & Caviglione, L. (2015). A survey on energy-aware security mechanisms. Pervasive and Mobile Computing, 24, 77–90. https://doi.org/10.1016/j.pmcj.2015.05.005
9. Rosendo, M., & Granjal, J. (2022). Energy-Aware Security Adaptation for Low-Power IoT Applications. Network, 2(1), 36–52. https://doi.org/10.3390/network2010003
10. Sun, X., Lian, W., Duan, H., & Wang, A. (2021). Factors affecting wind power efficiency: Evidence from provincial-level data in china. Sustainability (Switzerland), 13(22). https://doi.org/10.3390/su132212759
11. S, S. (2019). ENERGY-AWARE SECURITY ROUTING PROTOCOL FOR WSN IN BIG-DATA APPLICATIONS. Journal of ISMAC,

01(01), 39–55. https://doi.org/10.36548/jismac.2019.1.004

12. Song, X., Zhao, C., Han, J., Zhang, Q., Liu, J., & Chi, Y. (2020). Measurement and influencing factors research of the energy and power efficiency in China: Based on the supply-side structural reform perspective. Sustainability (Switzerland), 12(9). https://doi.org/10.3390/su12093879
13. Shaikh, A., Uddin, M., Elmagzoub, M. A., & Alghamdi, A. (2020). PEMC: Power efficiency measurement calculator to compute power efficiency and CO2emissions in cloud data centers. IEEE Access, 8, 195216–195228. https://doi.org/10.1109/ACCESS.2020.3033791