People's Democratic Republic of Algeria
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--T L E M C E N--



المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

A Thesis Presented in Partial Fulfillment of the Requirements for

The Degree of Master

Department: 2nd Cycle Studies

Specialisation: Electrical Engineering

Thesis Title

Event-Based Machine Learning in NILM: A Hardware-in-the-Loop Approach

Submitted by :

Abdelaziz Sami Khaldi

Date: 23rd June, 2026

Before a jury composed of:

Mrs. Kherbouche Djamila	MCA	ESSA. Tlemcen	President
Mr. Oujamaa Farid	MCB	ESSA. Tlemcen	Supervisor
Mr. Messirdi Mohammed	MCB	Univ. Mascara	Co-Supervisor
Mr Tahour Ahmed	Prof.	ESSA. Tlemcen	Examiner

Academic Year: 2025 / 2026

Dedication

To my mother and father,
for your unconditional love, endless sacrifices, and unwavering support throughout every stage of my life.
Your guidance, patience, and belief in me gave me the strength to persevere through challenges and pursue my ambitions.
This achievement is as much yours as it is mine.

To my grandparents,
whose wisdom, prayers, and affection have been a constant source of comfort and inspiration.
Your values and examples continue to guide me in ways words cannot fully express.

To my aunts, uncles, brothers, and cousins,
thank you for your encouragement, kindness, and the countless moments we have shared together.
Your presence has enriched my life and helped shape the person I have become.

And finally, to Oum Walid and Abu Walid from the Kingdom of Saudi Arabia,
and to your children, Walid, Abdellah, and especially Abdulaziz after whom I was named.
Though many years have passed, your kindness and care during my earliest childhood have never been forgotten.
Your boy has grown up and is now about to graduate.
I carry that connection with gratitude and pride.

To my friends,
for your companionship, shared laughter, and support throughout this journey.
Thank you for always being there to lift my spirits, share the challenges, and make these university years truly unforgettable.

To my cat, Brownie,
for your silent companionship, comforting presence, and for keeping me company during the countless late nights of study and drafting.

KHALDI Abdelaziz Sami

Acknowledgement

I would like to express my sincere gratitude to my thesis supervisor, Mr. Farid OUDJAMA, for his guidance, encouragement, and support throughout the realization of this work. I am particularly thankful for his openness and confidence in accepting the thesis topic that I proposed, allowing me the freedom to explore a challenging and innovative subject that aligned with my interests and aspirations.

I would also like to extend my appreciation to my co-supervisor, Mr. Mohammed Messirdi, for his valuable advice, constructive feedback, and continuous support during the development of this project.

My sincere thanks go to the members of the jury for dedicating their time and expertise to evaluate this work. I am especially grateful to Mrs. Kherbouche Djamila for presiding over the jury as President, and to Mr. Tahour Ahmed for serving as Examiner. Their effort and consideration are greatly appreciated.

I would like to acknowledge the Higher School of Applied Sciences of Tlemcen (ESSAT) for providing a stimulating academic environment and the resources necessary to carry out this study. I extend my heartfelt thanks to all the faculty members, teachers, and administration staff of ESSAT for their valuable instruction, constant support, and dedication throughout my academic journey.

Special thanks are extended to Ramzi and Abdelatif from the FABLAB laboratory, whose technical assistance, practical support, and willingness to help were invaluable throughout the implementation and testing phases of this project.

I would also like to express my gratitude to Mr. ROUABAH Abdeldjalil for his generosity and support in providing access to advanced technological resources that proved beneficial during the development and research stages of this work.

I would like to express my heartfelt gratitude to my brother, **KHALDI Mustapha Kamel**, for his invaluable advice, insightful guidance, and unwavering support throughout the writing of this thesis. His encouragement, constructive suggestions, and constant belief in my abilities were a source of motivation during every stage of this work. I am deeply grateful for his presence and assistance, which greatly contributed to the successful completion of this thesis.

Finally, I would like to thank all those who contributed, directly or indirectly,

to the completion of this thesis. Your support, assistance, and encouragement have played an important role in making this achievement possible.

الملخص

يعتمد التحول نحو الإدارة الذكية لجانب الطلب (DSM) بشكل كبير على الفصل الدقيق لاستهلاك الطاقة على مستوى الأجهزة. وفي حين أن المراقبة غير التداخلية للأحمال (NILM) المستمرة القائمة على السحابة والتي تستفيد من التعلم العميق توفر دقة نظرية عالية، إلا أنها تفرض قيوداً شديدة على زمن الاستجابة وعرض النطاق الترددي، بالإضافة إلى ثغرات في خصوصية البيانات. تقترح هذه المذكرة تحولاً جذرياً نحو منهجية تعتمد على الأحداث وحوسبة الحافة، مع تفصيل تصميم ونشر "نسناس" (Nesnas) — وهي أداة مستقلة تماماً لمراقبة الطاقة قادرة على تحديد الأجهزة المنزلية باستخدام التعلم الآلي المحلي.

يدمج "نسناس" واجهة تناظرية (AFE) مخصصة ومحولاً تناظرياً رقمياً بدقة 24 بت (ADS1256) لالتقاط بيانات القياس عن بُعد للشبكة المتوافقة طورياً بمعدل 1,600 عينة في الثانية .(SPS) ولتجاوز اختناقات المعالجة، تقوم بنية برمجية متعددة المسارات بعزل استعلام الأجهزة عن الاستدلال عالي المستوى. تُخزن مساحة الميزات الزمنية المستخرجة — والتي تشمل الطاقة الفعالة (P)، والمصفوفات التوافقية عبر تحويل فوريه السريع ،(FFT) والمقاييس الثابتة للشبكة — في نافذة زمنية منزلقة حتمية مدتها 3 ثوانٍ. وتحت إدارة آلة حالات محدودة (FSM) محلية، تعمل هذه الهيكلية كبوابة ذكية، حيث تقوم بتشغيل مصنف الغابة العشوائية Forest Random خفيف الوزن حسابياً لتحديد الجهاز المعين فقط عند اكتشاف حالات انتقالية فيزيائية واستقرارها.

يُظهر التحقق التجريبي باستخدام أحمال حثية متعددة الحالات مثل مجفف الشعر مقابل أحمال مقاومة بحتة مثل الغلاية أن نموذج التعلم الآلي المحلي يحدد الأجهزة بدقة وينجح في حل خطأ "التلطيخ المنزلق" (sliding smear) المتأصل في المراقبة المستمرة للطاقة. تُثبت معمارية "نسناس" أنه يمكن تحقيق تصنيف حتمي وبدون زمن انتقال لنظام (NILM) بالكامل عند الحافة، مما يوفر حلاً آمناً للغاية ويحافظ على الخصوصية لبيئات الشبكة الذكية الحديثة.

الكلمات المفتاحية: المراقبة غير التداخلية للأحمال ،(NILM) ذكاء حوسبة الحافة ،(Edge-AI) التعلم الآلي، الغابة العشوائية، التعرف على الأجهزة، معالجة الإشارات الرقمية.

Abstract

The transition toward intelligent Demand-Side Management (DSM) relies heavily on accurate, appliance-level energy disaggregation. While continuous, cloud-based Non-Intrusive Load Monitoring (NILM) leveraging Deep Learning offers high theoretical accuracy, it introduces severe latency, bandwidth constraints, and data privacy vulnerabilities. This thesis proposes a paradigm shift toward an event-based, edge-computing methodology, detailing the design and deployment of "Nesnas"—a fully autonomous Edge-AI energy-monitoring instrument capable of identifying household appliances using localized machine learning.

Nesnas integrates custom data-acquisition hardware to capture high-precision grid telemetry. To bypass processing bottlenecks, a multi-threaded software architecture isolates hardware polling from high-level inference. The system extracts key electrical features and buffers them into a deterministic temporal sliding window. Governed by a localized Finite State Machine (FSM), this architecture acts as an intelligent gatekeeper, triggering a computationally lightweight Random Forest classifier to identify the specific appliance only upon detecting physical state transitions.

Empirical validation using multi-state inductive loads (e.g., hair dryers) against purely resistive loads (e.g., kettles) demonstrates that the localized machine learning model accurately identifies appliances and resolves classification errors inherent to continuous energy monitoring. The Nesnas architecture proves that deterministic, zero-latency NILM classification can be achieved entirely at the edge, offering a highly secure, privacy-preserving solution for modern Smart Grid environments.

Keywords: NILM, Edge-AI, Machine Learning, Random Forest, Appliance Identification, Digital Signal Processing.

Résumé

La transition vers une gestion intelligente de la demande (DSM) repose fortement sur une désagrégation précise de l'énergie au niveau des appareils. Bien que l'Analyse Non Intrusive des Charges (NILM) continue et basée sur le cloud utilisant l'apprentissage profond (Deep Learning) offre une grande précision théorique, elle introduit une latence sévère, des contraintes de bande passante et des vulnérabilités liées à la confidentialité des données. Cette thèse propose un changement de paradigme vers une méthodologie basée sur les événements et l'informatique en périphérie (edge-computing), en détaillant la conception et le déploiement de « Nesnas » — un instrument de surveillance de l'énergie Edge-AI entièrement autonome, capable d'identifier les appareils électroménagers en utilisant un apprentissage automatique (machine learning) localisé.

Nesnas intègre un matériel d'acquisition de données sur mesure pour capturer une télémétrie de réseau de haute précision. Pour contourner les goulots d'étranglement de traitement, une architecture logicielle multithread isole l'interrogation du matériel de l'inférence de haut niveau. Le système extrait les principales caractéristiques électriques et les met en mémoire tampon dans une fenêtre glissante temporelle déterministe. Régie par une machine à états finis (FSM) localisée, cette architecture agit comme un contrôleur d'accès intelligent, déclenchant un classificateur de type forêt aléatoire (Random Forest) léger en calcul pour identifier l'appareil spécifique uniquement lors de la détection de transitions d'état physiques.

La validation empirique utilisant des charges inductives multi-états (par ex., sèche-cheveux) par rapport à des charges purement résistives (par ex., bouilloires) démontre que le modèle d'apprentissage automatique localisé identifie avec précision les appareils et résout les erreurs de classification inhérentes à la surveillance continue de l'énergie. L'architecture Nesnas prouve qu'une classification NILM déterministe et sans latence peut être réalisée entièrement à la périphérie, offrant une solution hautement sécurisée et respectueuse de la vie privée pour les environnements modernes de réseaux électriques intelligents (Smart Grid).

Mots-clés : NILM, Edge-AI, Apprentissage automatique, Forêt aléatoire, Identification d'appareils, Traitement du signal numérique.

Contents

List of Figures

List of Tables

Abreviations list

Introduction **1**

1 Foundations of Non-Intrusive Load Monitoring **3**
1.1 Introduction . . . 3
1.2 Background on Non-Intrusive Load Monitoring . . . 4
1.2.1 Historical Background . . . 4
1.2.2 NILM Principle . . . 5
1.3 NILM versus Intrusive Load Monitoring . . . 5
1.4 Appliance Signatures and Appliance Categories . . . 6
1.4.1 Appliance Signatures . . . 6
1.4.2 Appliance Categories . . . 7
1.5 NILM Processing Stages . . . 8
1.6 NILM Approaches . . . 8
1.7 Machine Learning in NILM . . . 9
1.8 Related Work . . . 9
1.9 Thesis Contribution and Scope . . . 12
1.10 Chapter Summary . . . 14

2 Hardware Architecture and Component Selection **15**
2.1 Introduction . . . 15
2.2 System Architecture Overview . . . 15
2.3 Voltage and Current Measurement Subsystem . . . 16
2.3.1 Voltage Measurement Using the ZMPT101B Sensor . . . 16
2.3.2 Current Measurement Using SCT-013 Split-Core Transformer . 18
2.4 Analog-to-Digital Conversion Using the ADS1256 . . . 20
2.5 The Human-Machine Interface and Bus Allocation . . . 21
2.5.1 Display Selection and Bus Contention . . . 21
2.5.2 Tactile Navigation Array . . . 22
2.6 Analog Front-End Design . . . 23
2.7 Chapter Summary . . . 25

3 Non-Intrusive Energy Signature Network for Appliance State-Identification (NESNAS) Realization **26**
3.1 Introduction . . . 26
3.2 Embedded Computing Platform . . . 26
3.2.1 Raspberry Pi Hardware . . . 26
3.2.2 Nesnas Operating System . . . 27
3.2.3 Remote System Administration . . . 27
3.3 Mechanical Design and Physical Realization . . . 28
3.3.1 Current Sensor Interface and Connector Mapping . . . 30
3.3.2 ADC Integration and Analog Pathways . . . 31
3.3.3 Digital Communication Interfaces . . . 32
3.3.4 Power Distribution and Grounding . . . 33
3.3.5 System Wiring and GPIO Allocation . . . 34
3.3.6 Sensor Integration and Mains Isolation . . . 35
3.4 Software Architecture . . . 37
3.4.1 Data Acquisition Engine . . . 38
3.4.2 Multithreaded Processing Architecture . . . 38
3.4.3 Data Routing and System Services . . . 39
3.4.4 Local User Interface . . . 39
3.4.5 Remote Monitoring Dashboard . . . 40
3.5 Feature Extraction Pipeline . . . 42
3.5.1 Signal Preprocessing . . . 42
3.5.2 Time-Domain Feature Computation . . . 43
3.5.3 Frequency and Harmonic Analysis . . . 44
3.5.4 Feature Vector Construction . . . 44
3.6 Local Human-Machine Interface (HMI) and Edge Navigation . . . 44
3.7 Chapter Summary . . . 47

4 Experimental Evaluation and Appliance Classification **48**
4.1 Introduction . . . 48
4.2 Machine Learning Background . . . 48
4.2.1 Supervised Learning . . . 49
4.2.2 Random Forest . . . 49
4.2.3 Comparative Classification Models . . . 50
4.3 Experimental Methodology . . . 50
4.3.1 Test Bench . . . 51
4.3.2 Safety and Electrical Isolation . . . 52
4.3.3 Appliance Selection . . . 52
4.3.4 Acquisition Procedure . . . 52
4.3.5 Data Validation and Benchmarking . . . 54
4.4 Experimental Dataset . . . 60
4.4.1 Dataset Preparation . . . 60
4.4.2 Dataset Statistics . . . 61
4.5 Feature Engineering . . . 62
4.5.1 Electrical Feature Extraction . . . 62
4.5.2 Temporal Window Generation . . . 63
4.5.3 Data Augmentation . . . 63
4.5.4 Training and Testing Dataset Generation . . . 63

4.6 Model Training 64
4.6.1 Validation Strategy 65
4.7 Experimental Results 66
4.7.1 Classifier Performance Comparison 66
4.7.2 Confusion Matrix Analysis 66
4.8 Real-Time Inference Framework 67
4.8.1 Live Deployment Validation 68
4.9 Chapter Summary 70

General Conclusion and Future Perspectives **71**

Appendix **74**

A Mechanical CAD Evolution and Enclosure Iterations **74**

B Software Architecture and Execution Logic **77**
B.1 The `NesnasCoreEngine`: DSP Pipeline 77
B.2 The Temporal Sliding-Window and Inference FSM 79

List of Figures

1.1 Side-by-side architectural comparison between ILM and NILM deployments. . . . 6

2.1 Active voltage transformer module (ZMPT101B), Image source: [8]. . . 17
2.2 External connection diagram of the ZMPT101B module, Adapted from [8]. . . . 17
2.3 Internal schematic of the ZMPT101B active module, Source: [8]. . . . 17
2.4 Split-core current transformer (SCT-013-030) [27]. . . . 18
2.5 Standard DC bias conditioning circuit Schematic adapted from [18]. . . 19
2.6 The 24-bit ADC module (ADS1256) [25]. . . . 20
2.7 Front view of the 2.42-inch OLED [23]. . . . 22
2.8 Rear view detailing the isolated I^2C interface [23]. . . . 22
2.9 Ideal monolithic 4-button array for interface navigation [17]. . . . 22
2.10 Implemented single-button logic module utilized via hardware interrupts [17]. . . . 22
2.11 Schematic of the custom Analog Front-End (AFE). . . . 23
2.12 GERBER layout of the AFE designed in EasyEDA. . . . 24
2.13 3D render of the AFE component side. . . . 25
2.14 3D render of the AFE solder side. . . . 25
2.15 Physical realization of the AFE prototype on perfboard: front view showing component placement. . . . 25
2.16 Physical realization of the AFE prototype on perfboard: back view showing hand-soldered routing. . . . 25

3.1 The Tailscale administrative dashboard, displaying the active connection to the `nesnas` Edge node. . . . 28
3.2 Architectural representation of the remote development environment. . 28
3.3 Front and top view of the fabricated NESNAS instrument, featuring the local HMI and ventilation. . . . 30
3.4 Rear view demonstrating sensor integration, panel-mount audio jacks, and cable management. . . . 30
3.5 Anatomy of the 3.5 mm audio jack used to anchor current telemetry to the custom AFE. [7]. . . . 31
3.6 Routing topology of the AFE perfboard. . . . 33
3.7 Physical pin mapping of the Raspberry Pi 4 GPIO header used in the NESNAS instrument. [11]. . . . 35
3.8 Comparison of conductor termination practices at the ZMPT101B mains-voltage sensing interface. . . . 36
3.9 Installation protocol for the SCT-013 Current Transformer. . . . 37

3.10 Bifurcated software architecture. 39
3.11 Macroscopic overview of the developed web dashboard interface for the Air Conditioner node. 41
3.12 Detailed metrology analysis panels for the Air Conditioner node: left panel view showing real-time power metrics and waveform shapes. . . . 41
3.13 Detailed metrology analysis panels for the Air Conditioner node: right panel view showing harmonic distribution array and logging. 42
3.14 CAD layout of the front panel HMI. 45
3.15 Button 1 logic displaying the live metrology dashboard. 46
3.16 Button 2 logic demonstrating the system entering ML Studio mode. . . 46
3.17 Button 3 logic indicating the activation of the physical recording sequence. 46
3.18 Button 4 logic initiating a secure, programmatic termination of the Linux kernel to protect the file system. 47

4.1 Conceptual illustration of the Random Forest classification process. [1]. 50
4.2 Experimental test bench setup. 51
4.3 Nesnas ML Studio interface during appliance event acquisition. 53
4.4 Termination of the acquisition session after completion of the recording sequence. 53
4.5 Experimental setup used during the benchmarking and validation experiments. 54
4.6 Macroscopic overview of the web dashboard for the Kettle node. 55
4.8 Localized Nesnas OS OLED dashboard during the kettle test. 55
4.7 Detailed metrology analysis panels for the Kettle node. 56
4.9 Macroscopic overview web dashboard interface for the Hairdryer 57
4.10 Detailed metrology analysis panels for the Hairdryer. 58
4.11 Localized Nesnas OS OLED dashboard during the hairdryer test. . . . 59
4.12 Fluke 179 True RMS Multimeter acting as the empirical witness. . . . 59
4.13 Fluke 179 True RMS Multimeter acting as the empirical witness. . . . 60
4.14 Distribution of samples across the seven appliance classes used for machine-learning training and evaluation. 62
4.15 Cyclic machine learning workflow. 64
4.16 Offline training output obtained on the workstation after training the Random Forest classifier on the augmented dataset. 65
4.17 Confusion matrix of the Random Forest classifier evaluated on the held-out test set. 67
4.18 Live appliance classification performed by the deployed Edge-AI system using the temporal sliding-window inference framework. 69

A.1 Mechanical enclosure designs developed for the NESNAS platform. Figures (1) and (2) present preliminary CAD design concepts, while Figures (3) and (4) show the two fabricated enclosure prototypes used during system development. 74
A.2 Left-side view of Prototype 1, illustrating the side-mounted component topology. 75
A.3 Right-side view of Prototype 1, demonstrating hazardous strain on internal wiring upon opening. 75
A.4 CAD rendering of Prototype 2, demonstrating the modular six-piece architecture designed to eliminate wiring strain. 75

A.5 Internal CAD topology of Prototype 2. Components are anchored safely: Raspberry Pi 4 (1), active cooling fan (2), ADS1256 (3), ZMPT101B (4), AFE board (5), chassis handles (6), and vented top cap (7). 76
A.6 2D DXF profile generated for the transparent diagnostic window. . . . 76
A.7 CAD rendering of the rear chassis cap featuring the integrated NESNAS branding. 76

B.1 Execution logic for the high-priority hardware polling and DSP thread. This loop strictly avoids blocking I/O to maintain the 1800 SPS empirical limit. 78
B.2 Execution logic for the Inference Finite State Machine. The 3-second sliding window prevents the classifier from triggering false positives during physical appliance actuation. 79

List of Tables

1.1 Comparison of steady-state and transient appliance signatures commonly used in Non-Intrusive Load Monitoring (NILM). Compiled from [30, 14, 21]. 7

2.1 Critical Metrology Specifications of the ZMPT101B Transformer 18
2.2 Critical Metrology Specifications of the SCT-013-030 Transformer . . . 19
2.3 Critical Digital Conversion Specifications of the ADS1256 21

3.1 Key Specifications of the Raspberry Pi 4 Model B 27
3.2 Thermal Performance of the Raspberry Pi 4 Under Continuous Inference Load 29
3.3 Analog Metrology Pathways (Sensors to AFE to ADS1256) 31
3.4 Digital Communication (SPI, I2C & HMI Buttons) 32
3.5 Power Distribution and Ground Rails (Perfboard Routing) 34
3.6 High-Voltage Mains Integration Pathways 37

4.1 Excerpt from the exported CSV dataset 54
4.2 Comparison between the proposed measurement subsystem and the Fluke 179 True RMS reference instrument during benchmarking experiments. 60
4.3 Dataset size and class distribution. 61
4.4 Cross-validation performance of evaluated classification models. 66

Abreviations list

AC	Alternating Current
ADC	Analog-to-Digital Converter
AFE	Analog Front-End
AI	Artificial Intelligence
CAD	Computer-Aided Design
CPU	Central Processing Unit
CSV	Comma Separated Values
DC	Direct Current
DL	Deep Learning
DSM	Demand-Side Management
DSP	Digital Signal Processing
EMI	Electromagnetic Interference
FFT	Fast Fourier Transform
FSM	Finite State Machine
GIL	Global Interpreter Lock
GPIO	General-Purpose Input/Output
HMI	Human-Machine Interface
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
ILM	Intrusive Load Monitoring
LSTM	Long Short-Term Memory
ML	Machine Learning
NESNAS	Non-Intrusive Energy Signature Network for Appliance State-Identification
NILM	Non-Intrusive Load Monitoring
OLED	Organic Light-Emitting Diode
OS	Operating System
PCB	Printed Circuit Board
RMS	Root Mean Square
SD	Secure Digital

SPI	Serial Peripheral Interface
SPS	Samples Per Second
SSH	Secure Shell
THD	Total Harmonic Distortion
TPU	Tensor Processing Unit
VPN	Virtual Private Network

Introduction

The evolution of conventional power networks into data-driven Smart Grids has fundamentally reshaped energy management by enabling real-time monitoring, consumer participation, and adaptive control mechanisms. At the core of this transition lies Demand-Side Management (Demand-Side Management (DSM)), a framework designed to empower consumers and utility providers to dynamically monitor, optimize, and minimize energy consumption. However, the efficacy of DSM is fundamentally constrained by measurement resolution, since effective optimization of residential energy consumption requires detailed visibility into the operation of individual appliances. Traditionally, this level of observability has been achieved through Intrusive Load Monitoring (Intrusive Load Monitoring (ILM)), which employs dedicated sub-meters to monitor each appliance separately. Despite its high accuracy, ILM suffers from significant practical limitations, including high installation costs, intrusive hardware requirements, and limited scalability.

To overcome the practical limitations of ILM, Non-Intrusive Load Monitoring (NILM) was introduced as an alternative approach for appliance-level energy monitoring. By collecting electrical measurements, such as instantaneous voltage $v(t)$ and current $i(t)$, at a single point of observation near the main circuit breaker, NILM uses signal processing and pattern recognition methods to infer the operating states and energy consumption of individual appliances.

As NILM research matured, improving disaggregation accuracy became a primary objective, leading to the adoption of deep learning techniques for appliance identification and load disaggregation. Among these approaches, Long Short-Term Memory (Long Short-Term Memory (LSTM)) networks have received considerable attention due to their ability to capture temporal dependencies in sequential electrical measurements. However, the computational complexity of deep learning models often necessitates the transmission of measurement data to remote cloud servers for processing. This reliance on cloud-based inference introduces several practical challenges, including increased bandwidth consumption, potential privacy risks associated with transmitting household energy data, and communication latency. Such limitations can compromise the near real-time operation required by advanced energy management and Smart Grid applications.

To address the limitations associated with cloud-based NILM deployment, this thesis investigates an edge-computing approach that brings data acquisition and inference closer to the point of measurement. Specifically, it presents the design, implementation, and experimental evaluation of NESNAS, a Hardware-in-the-Loop Edge-Artificial Intelligence (AI) platform for autonomous appliance-level energy monitoring. By integrating electrical measurement and data acquisition, embedded processing, and local inference within a single system, the proposed platform aims to reduce communication latency, mitigate privacy concerns, and enable real-time load disaggregation

without reliance on remote cloud infrastructure.

At the hardware level, Nesnas combines a custom AFE with a phase-coherent 24-bit Analog-to-Digital Converter (ADC) (ADS1256) to acquire high-fidelity electrical measurements at 1,600 Samples Per Second (SPS). These measurements are processed by a single-board computing platform running a multi-threaded software architecture that coordinates acquisition, processing, and inference in real time.

Building upon this hardware platform, a key contribution of this thesis is the development of an event-based machine learning framework for load identification. Rather than performing continuous classification on the incoming measurement stream, the proposed approach first extracts appliance signatures through a dedicated Digital Signal Processing (DSP) pipeline. Real-time Active Power (P), Reactive Power (Q), and harmonic components obtained from Fast Fourier Transform (FFT) analysis are combined to form a feature vector $\mathbf{x}(t)$ that characterizes appliance operating events. These features are subsequently processed by a computationally efficient Random Forest classifier, making the approach suitable for resource-constrained edge devices. To improve classification stability, the inference process is governed by a 3-second temporal sliding window and a localized Finite State Machine (FSM), which together reduce transient misclassifications and enhance the discrimination of appliances exhibiting distinct electrical behaviors, ranging from binary resistive loads such as electric kettles to multi-state inductive devices such as hairdryers.

The reminder of this thesis is organized as follows:

Chapter 1: NILM Generalities and State of the Art presents the theoretical foundations of energy disaggregation and reviews the evolution of NILM methodologies. It examines the limitations of conventional cloud-based architectures and introduces the motivation for localized real-time load monitoring and event-based classification approaches to load monitoring.

Chapter 2: Hardware Architecture and Component Selection describes the design and implementation of the measurement subsystem. It discusses the selection of sensing and data-acquisition components and presents the signal-conditioning techniques employed to ensure reliable acquisition of electrical and harmonic measurements.

Chapter 3: Prototype Realization and Embedded Feature Extraction details the development of the Nesnas platform and its embedded software architecture. It presents the real-time data-acquisition framework and the DSP procedures used to extract electrical features and construct the appliance characterization vector.

Chapter 4: Edge-AI Architecture and Random Forest Classification introduces the proposed event-based classification framework. It describes the feature-space formulation, the integration of the temporal sliding window and the localized FSM, and the implementation of the Random Forest classifier. The chapter concludes with an experimental evaluation of the proposed system under representative operating conditions.

General Conclusion and Future Perspectives summarizes the main contributions of this work, discusses its limitations, and outlines potential directions for future research and system development.

Chapter 1

Foundations of Non-Intrusive Load Monitoring

1.1 Introduction

The continuous growth in global energy demand has become one of the major challenges facing modern societies. Rapid population growth, urbanization, industrial development, and the increasing adoption of electrical technologies have significantly increased electricity consumption worldwide. At the same time, concerns regarding environmental sustainability, energy efficiency, and greenhouse gas emissions have encouraged governments, utility companies, and researchers to develop more effective energy management solutions [30, 21, 4].

The residential sector represents a substantial portion of total electricity consumption and offers significant opportunities for energy savings. Improving energy efficiency within households requires a detailed understanding of how electrical energy is consumed by individual appliances. Such information enables consumers to identify energy-intensive devices, detect abnormal operating conditions, and adopt more efficient consumption habits. Furthermore, appliance-level monitoring can support demand-side management strategies, load forecasting, fault detection, and smart home automation systems [30, 21].

The deployment of smart grids and advanced metering infrastructures has improved the ability to monitor electrical energy consumption. Modern smart meters provide detailed information regarding aggregate household energy usage and facilitate two-way communication between consumers and utility providers. Despite these advancements, conventional energy meters typically report only the total energy consumed by a household and do not provide information regarding the contribution of individual appliances [30, 21].

Obtaining appliance-level energy information using traditional monitoring approaches generally requires the installation of dedicated sensors on each appliance, a technique commonly referred to as Intrusive Load Monitoring (ILM). Although ILM can provide highly accurate measurements, its practical deployment is often limited by increased hardware costs, installation complexity, maintenance requirements, and scalability challenges. As the number of monitored appliances increases, the cost and complexity of the monitoring system grow significantly [30, 21].

To overcome these limitations, Non-Intrusive Load Monitoring (NILM) has emerged as an attractive alternative for appliance-level energy monitoring. NILM aims

to estimate the operating states and energy consumption of individual appliances using measurements acquired from a single monitoring location, typically at the main electrical panel or smart meter. By analyzing aggregate electrical signals, NILM eliminates the need for dedicated sensors on every appliance while maintaining the capability to provide detailed consumption information [9, 30, 14, 21].

A significant challenge in NILM research is balancing identification accuracy with implementation complexity. Many modern NILM approaches rely on high-frequency electrical measurements and sophisticated signal processing techniques to capture transient appliance signatures. Although these methods often achieve high classification performance, they require expensive acquisition hardware, large storage capacities, and substantial computational resources. Such requirements can limit their practical deployment in residential environments and low-cost monitoring systems [30, 14, 21]

Event-based NILM has attracted considerable attention as a practical alternative to high-frequency approaches. Rather than continuously processing large volumes of electrical data, event-based methods focus on detecting appliance switching events and extracting characteristic features associated with state transitions. This approach reduces storage requirements, communication bandwidth, and computational complexity while maintaining effective appliance identification capabilities. Furthermore, event-based NILM systems are particularly suitable for implementation on embedded platforms and real-time monitoring applications [9, 30, 4]

These considerations motivate the development of practical NILM solutions capable of providing reliable appliance identification while maintaining low hardware and computational requirements. In this context, event-based monitoring combined with machine learning techniques represents a promising approach for achieving accurate and scalable appliance-level energy monitoring in residential environments [21, 4].

In addition to algorithmic challenges, practical NILM deployment requires suitable data acquisition hardware capable of providing reliable electrical measurements at low cost. Existing commercial smart meters are primarily designed for billing and energy reporting purposes and often provide limited access to the electrical measurements required for NILM research and appliance identification. Consequently, the development of dedicated acquisition platforms remains an important aspect of practical NILM system design. This consideration motivates the development of the NESNAS (Non-Intrusive Energy Signature Network for Appliance State-Identification) platform presented in this thesis, which was designed to support event-based appliance monitoring and machine-learning-based load identification.

1.2 Background on Non-Intrusive Load Monitoring

1.2.1 Historical Background

Non-Intrusive Load Monitoring (NILM), also known as energy disaggregation, was first introduced by Hart in 1992 as a method for estimating the energy consumption of individual electrical appliances from measurements obtained at a single monitoring point [9]. The fundamental objective of NILM is to determine the operating states and energy consumption of individual appliances by analyzing aggregate electrical measurements collected from the main supply of a building or household. Hart demonstrated that changes in aggregate power consumption contain distinctive appliance signatures that can be used to identify individual devices and track their operation over time [9].

Since its introduction, NILM has evolved into a multidisciplinary research field involving power systems, signal processing, machine learning, embedded systems, and smart grid technologies. The widespread deployment of smart meters and the increasing availability of computational resources have significantly accelerated research in NILM, enabling the development of more sophisticated load disaggregation algorithms and real-time monitoring solutions [30, 14, 21, 4]. Today, NILM is considered a key enabling technology for appliance-level energy monitoring, demand-side management, fault detection, energy auditing, and smart home automation applications [21, 4].

1.2.2 NILM Principle

The operating principle of NILM is based on the observation that the total electrical power consumed by a household is equal to the sum of the power consumed by all operating appliances [9]. The aggregate power measured at the main electrical panel can be expressed as

$$P_{\text{total}}(t) = \sum_{i=1}^{N} P_i(t) \tag{1.1}$$

where $P_{\text{total}}(t)$ denotes the aggregate power consumption, $P_i(t)$ represents the power consumption of appliance i, and N is the total number of appliances operating at time t.

In practical implementations, the aggregate measurement also contains noise and contributions from unidentified devices. Therefore, the NILM problem is commonly represented as [21, 4]

$$P_{\text{agg}} = \sum_{i=1}^{N} P_i + e \tag{1.2}$$

where e represents measurement noise and unknown electrical loads.

The objective of NILM is to estimate the contribution of each appliance from the aggregate measurement. This task is challenging because multiple appliances may operate simultaneously, resulting in overlapping electrical signatures and complex power consumption patterns [30, 14].

1.3 NILM versus Intrusive Load Monitoring

Appliance-level energy monitoring can be achieved using either Intrusive Load Monitoring (ILM) or Non-Intrusive Load Monitoring (NILM) [30, 21].

In ILM systems, individual sensors or smart plugs are installed on each appliance to directly measure its electrical consumption. Although this approach provides highly accurate appliance-level information, it requires substantial hardware investment, extensive installation effort, and increased maintenance complexity. Furthermore, the addition of new appliances typically requires the deployment of additional sensors [30].

In contrast, NILM utilizes measurements acquired from a single monitoring point, such as a smart meter installed at the household entry point, to estimate the operation and energy consumption of multiple appliances simultaneously. This significantly reduces installation costs and simplifies deployment while maintaining the ability to provide appliance-level information [9, 21].

Consequently, NILM has become the preferred solution for large-scale residential energy monitoring due to its scalability, flexibility, and compatibility with existing smart-metering infrastructures [21, 4]. The conceptual difference between ILM and NILM is illustrated in Figure 1.1.

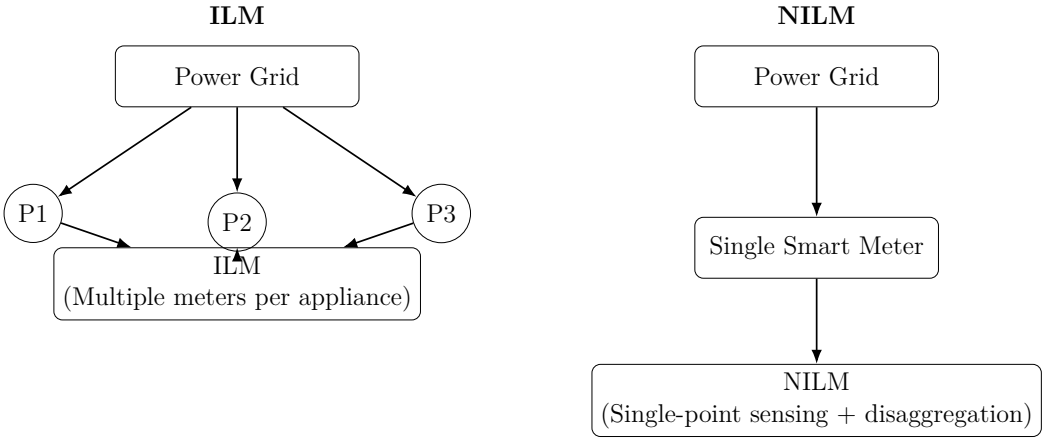


Figure 1.1: Side-by-side architectural comparison between ILM and NILM deployments.

1.4 Appliance Signatures and Appliance Categories

1.4.1 Appliance Signatures

A fundamental concept in NILM is the appliance signature, which refers to a set of electrical characteristics that uniquely describe the behavior of an appliance during operation [9, 30]. Since different appliances exhibit distinct electrical characteristics, these signatures can be used to identify and distinguish appliances from one another.

Appliance signatures are generally classified into steady-state and transient-state signatures [9, 14]. Steady-state signatures are extracted when an appliance operates in a stable state and commonly include:

- Active power (P)
- Reactive power (Q)
- Apparent power (S)
- Current magnitude (I)
- Voltage magnitude (V)
- Power factor (PF)
- Harmonic components

Transient signatures are extracted during appliance switching events and may include transient power, current spikes, waveform shape, transient duration, and harmonic content [30].

Among these features, changes in active and reactive power are the most widely used signatures in event-based NILM systems [9, 30]. These features can be calculated as

$$\Delta P = P_{\text{after}} - P_{\text{before}} \tag{1.3}$$

and

$$\Delta Q = Q_{\text{after}} - Q_{\text{before}} \tag{1.4}$$

where the subscripts *before* and *after* correspond to the steady-state conditions immediately surrounding a switching event.

The characteristics of commonly used NILM feature categories can be summarized according to their sampling requirements and the type of information they provide for appliance identification, as shown in Table 1.1 [30, 14, 21].

Table 1.1: Comparison of steady-state and transient appliance signatures commonly used in NILM. Compiled from [30, 14, 21].

Category	**Typical Features**	**Sampling Rate Req.**	**Primary Advantage**
Steady-State (Low-Frequency)	Active Power (P), Reactive Power (Q), Root Mean Square (RMS) Current	Low (< 1 Hz)	Low compute, efficient logging and transmission.
Transient (High-Frequency)	Harmonics (FFT), Total Harmonic Distortion (THD), Admittance (Y), Transients	High (> 1 kHz)	Highly discriminative for multi-state non-linear loads.

1.4.2 Appliance Categories

Electrical appliances can be classified according to their operating behavior and power consumption characteristics [9, 30, 21].

- **ON/OFF Appliances:** ON/OFF appliances operate using a single active state and are characterized by a distinct transition between OFF and ON conditions. Examples include electric kettles, lamps, toasters, and hair dryers [9].
- **Multi-State Appliances:** Multi-state appliances operate at several discrete power levels and can be represented using finite-state models. Examples include microwave ovens, washing machines, and air-conditioning systems [9, 30].
- **Continuously Variable Appliances:** Continuously variable appliances exhibit smooth and continuous changes in power consumption rather than discrete state transitions. Examples include inverter-driven air conditioners, variable-speed drives, and dimmable lighting systems [9, 21].

- **Non-Linear Appliances:** Modern electronic devices equipped with switched-mode power supplies often exhibit non-linear power consumption patterns characterized by significant fluctuations and harmonic distortions. Examples include laptops, televisions, and computer systems [21].

The existence of different appliance categories significantly increases the complexity of the NILM problem and influences the selection of appropriate feature extraction and classification techniques.

1.5 NILM Processing Stages

A typical NILM system consists of four main processing stages [30, 21].

- **Data Acquisition:** Electrical measurements are acquired at a single monitoring point using sensors and data acquisition hardware. Common measurements include voltage, current, active power, and reactive power.
- **Event Detection:** The purpose of event detection is to identify significant changes in electrical signals corresponding to appliance state transitions. Event detection reduces computational requirements by focusing subsequent processing only on relevant signal changes [9].
- **Feature Extraction:** After event detection, relevant electrical features are extracted from the measured signals. These features capture appliance-specific characteristics and serve as the basis for appliance identification.
- **Load Identification:** The final stage involves assigning each detected event to a specific appliance. This process is commonly performed using machine learning algorithms trained on labeled appliance signatures [30].

1.6 NILM Approaches

NILM methods can be broadly classified into three categories [21, 4].

- **Event-Based NILM:** Event-based methods identify appliance state transitions and extract features associated with switching events. These approaches typically require low sampling frequencies and exhibit low computational complexity [9].
- **State-based NILM:** State-based approaches analyze continuous consumption patterns without explicitly detecting appliance switching events. These methods often rely on probabilistic models, pattern recognition techniques, and machine learning algorithms [21].
- **High-Frequency NILM:** High-frequency approaches utilize transient waveform characteristics acquired using high-speed data acquisition systems. Although these methods can achieve high disaggregation accuracy, they require more expensive hardware and increased computational resources [30].

1.7 Machine Learning in NILM

Machine learning techniques have become increasingly important in NILM because of their ability to model complex relationships between electrical signatures and appliance identities [21, 4].

Commonly used machine learning algorithms include:

- Decision Trees
- k-Nearest Neighbors (kNN)
- Support Vector Machines (SVM)
- Random Forests
- Hidden Markov Models (HMM)
- Artificial Neural Networks (ANN)
- Deep Learning models such as CNNs and LSTMs

Among these approaches, Random Forest has become a popular choice for event-based NILM applications due to its robustness, ability to handle nonlinear feature relationships, resistance to overfitting, and efficient training and inference performance. Furthermore, Random Forest can operate effectively using low-frequency electrical signatures such as active and reactive power changes, making it suitable for practical smart-meter-based implementations [21, 4]. These characteristics make Random Forest particularly attractive for deployment in embedded and edge-based NILM systems where computational resources are limited.

1.8 Related Work

NILM was first introduced by Hart in 1992, who proposed the concept of identifying individual appliance operation from aggregated electrical measurements acquired at a single monitoring point. Hart's pioneering work established the fundamental principles of load disaggregation by employing active and reactive power measurements, event detection, clustering techniques, and finite-state appliance models. The proposed framework demonstrated that appliance operating states could be inferred from changes in aggregate power consumption without requiring dedicated sensors on each device. As a result, Hart's work became the foundation of modern NILM research and remains the basis of many contemporary disaggregation algorithms [9].

Following Hart's contribution, several studies sought to formalize NILM methodologies and evaluate their practical deployment. Zeifman and Roth [28] provided one of the earliest comprehensive reviews of NILM , discussing appliance signatures, event detection techniques, and disaggregation algorithms while identifying key challenges related to measurement accuracy, appliance diversity, and real-world implementation. Their review highlighted the growing importance of smart metering infrastructures and established a research roadmap that significantly influenced subsequent NILM research.

A major research focus during the early development of NILM was the extraction of discriminative appliance signatures. Researchers investigated steady-state characteristics such as active power, reactive power, power factor, and voltage-current relationships to distinguish individual appliances. Additional efforts explored transient signatures generated during appliance switching events, including inrush currents, transient power changes, harmonic distortions, and waveform characteristics. These transient features often provide greater discriminatory capability than steady-state measurements; however, they typically require high-frequency sampling rates and more sophisticated measurement hardware, increasing system complexity and implementation cost.

The work [30] presented a comprehensive survey of NILM methodologies and organized NILM systems into three principal stages: data acquisition, feature extraction, and inference and learning. Their work emphasized that event detection and appliance signature extraction are fundamental components of practical NILM systems. The authors categorized appliance signatures into steady-state, transient-state, and non-traditional features and highlighted active and reactive power as particularly attractive low-cost signatures because they can be obtained using conventional smart-meter hardware. Furthermore, the survey discussed the trade-off between low-frequency and high-frequency monitoring approaches, noting that while high-frequency measurements improve appliance discrimination through transient and harmonic analysis, they require expensive acquisition systems, increased storage capacity, and greater computational resources.

As NILM research matured, greater attention was given to appliance modeling and systematic categorization of disaggregation techniques. The authors in [14] presented a detailed review and taxonomy of NILM approaches, classifying appliances into on/off, multi-state, and infinite-state categories according to their operating behavior. The authors further categorized appliance signatures into steady-state and transient-state features and reviewed supervised and unsupervised learning methods used for appliance recognition. Their work emphasized that active and reactive power-based event detection remains one of the most practical approaches for real-world deployment due to its low hardware requirements and compatibility with conventional smart-meter sampling rates. The review also highlighted the importance of event-based NILM systems, noting that appliance state transitions can be effectively identified while maintaining relatively low computational complexity.

The increasing availability of publicly accessible datasets such as REDD, UK-DALE, AMPds, and GREEND facilitated the application of machine learning techniques to NILM problems. Traditional classification algorithms including k-Nearest Neighbours (KNN), Support Vector Machines (SVM), Decision Trees (DT), Bayesian classifiers, Random Forests (RF), and Hidden Markov Models (HMM) were widely investigated for appliance identification. Among probabilistic approaches, HMMs and Factorial Hidden Markov Models (FHMM) became popular because of their ability to model appliance state transitions and temporal dependencies. However, these methods often suffer from increasing computational complexity as the number of monitored appliances grows.

Several studies further investigated state-based and probabilistic appliance modeling. For example, the work in [6] proposed a NILM framework based on finite-state machines with fuzzy transitions to improve appliance state representation under uncertain operating conditions. Similarly, the work in [24] introduced a multi-label classification framework capable of simultaneously estimating the operating states of multiple

appliances from aggregate measurements. These approaches demonstrated the potential of machine learning techniques to improve appliance recognition while preserving interpretable state representations.

Event detection has emerged as a critical component of practical NILM systems because it reduces computational requirements by focusing processing resources on appliance state transitions rather than continuously analyzing the entire power stream. Various event detection approaches have been proposed, including threshold-based methods, transient analysis techniques, clustering algorithms, and machine-learning-based detectors. More recently, the work in [12] introduced the Multi-Environment Event Detector (MEED), an unsupervised framework capable of identifying appliance events across multiple operating environments without requiring extensive manual configuration. Event-based approaches remain particularly attractive for embedded NILM implementations because they reduce storage requirements, communication bandwidth, and computational complexity while maintaining effective appliance identification performance.

Recent NILM research has increasingly adopted machine learning and deep learning techniques. The authors in [13] introduced Neural NILM, demonstrating that deep neural networks could automatically learn appliance characteristics directly from aggregate measurements without extensive manual feature engineering. Their work represented one of the earliest successful applications of deep learning to energy disaggregation. Subsequently, the authors in [16] employed Long Short-Term Memory (LSTM) networks to capture temporal appliance behavior and improve disaggregation performance. Later, the authors in [29] proposed the Sequence-to-Point learning framework, which became one of the most influential deep-learning architectures for NILM by directly mapping aggregate power windows to appliance-level estimates.

Building upon these developments, researchers have investigated numerous deep learning architectures including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Bidirectional LSTM (BiLSTM), Gated Recurrent Units (GRU), denoising Autoencoders, Generative Adversarial Networks (GAN), and Transformer-based models. These approaches have demonstrated substantial improvements in appliance identification accuracy and energy disaggregation performance. However, their practical deployment often requires large labelled datasets, extensive training time, high computational power, and powerful hardware platforms, which may limit their suitability for low-cost embedded monitoring systems.

In [20] the authors provided an updated review of NILM research and proposed a taxonomy that groups disaggregation methods into machine learning, pattern matching, and source separation approaches. Their review highlighted the growing importance of smart-meter-based NILM systems and discussed the increasing use of deep learning architectures. The authors emphasized that machine learning approaches now dominate NILM research, while pattern matching techniques such as Dynamic Time Warping and source separation methods including Independent Component Analysis and Non-Negative Matrix Factorization continue to offer viable alternatives in specific applications.

An important observation made by [20] is that low-frequency active and reactive power measurements remain highly effective for appliance identification despite the increasing popularity of high-frequency waveform analysis. In subsequent work, Schirmer and Mporas [21] demonstrated that reliable energy disaggregation can be achieved using active and reactive power signatures alone, avoiding the hardware com-

plexity associated with transient and waveform-based approaches. Such low-frequency feature sets are particularly attractive for smart-meter-based and embedded NILM systems because they reduce sensing requirements, storage demands, and computational overhead while maintaining competitive classification performance.

More recently, the work in [4] reviewed online NILM techniques designed for real-time monitoring and industrial applications. Their work examined edge-computing and cloud-based implementations while highlighting challenges associated with scalability, computational complexity, communication requirements, and hardware limitations. The review demonstrated that machine learning techniques such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks are increasingly preferred for online NILM due to their robustness and computational efficiency. Furthermore, the authors emphasized that event-based approaches remain highly attractive for embedded and real-time implementations because they reduce processing requirements and communication overhead while preserving disaggregation accuracy.

Despite the significant progress achieved in NILM research over the past three decades, several challenges remain for the practical deployment of appliance-level energy monitoring systems. Many high-performance approaches rely on high-frequency measurements, waveform analysis, harmonic extraction, or deep learning architectures that require expensive acquisition hardware, large labelled datasets, and substantial computational resources [30, 21]. Although these methods can achieve excellent disaggregation performance, their implementation complexity often limits their suitability for low-cost residential monitoring applications and embedded platforms.

More recently, research has increasingly focused on online, edge-based, and real-time NILM implementations capable of processing electrical measurements locally while reducing communication overhead and dependence on cloud infrastructure [4]. These developments highlight the growing need for integrated solutions that combine efficient event detection, low-frequency electrical measurements, dedicated acquisition hardware, and computationally lightweight machine learning algorithms within a unified framework.

Despite these advances, comparatively limited attention has been devoted to the development of complete NILM systems that jointly address hardware acquisition, event-based processing, feature extraction, and appliance classification using low-cost and deployable architectures. Consequently, there remains a need for practical solutions capable of providing reliable appliance identification while maintaining low hardware requirements and computational complexity.

Motivated by these observations, this thesis develops the NESNAS acquisition platform and proposes an event-based appliance identification framework based on electrical feature extraction and Random Forest classification. The proposed approach aims to provide a practical and scalable solution for residential appliance monitoring using low-frequency measurements while supporting future real-time and edge-based NILM implementations.

1.9 Thesis Contribution and Scope

The scope of this thesis is limited to residential NILM applications based on event-driven appliance identification. The work focuses on the acquisition and processing of electrical measurements obtained from household appliances operating under controlled conditions.

Although NILM is often associated with full energy disaggregation, the present work focuses primarily on appliance identification through the classification of detected switching events rather than the complete reconstruction of appliance-level power consumption profiles. Consequently, the objective is to identify appliance operation from detected events rather than to estimate the continuous power consumption of all appliances over time.

Emphasis is placed on the development of a practical and low-cost monitoring framework capable of identifying appliance switching events and classifying appliances using machine learning techniques.

Within this scope, the primary contributions of this thesis are summarized as follows:

- **Design and Development of the NESNAS Platform**

  A dedicated hardware platform, referred to as NESNAS, was designed and implemented to acquire electrical measurements suitable for NILM applications. The platform provides a flexible and cost-effective solution for data acquisition and experimental evaluation.

- **Measurement Acquisition and Dataset Construction**

  Electrical measurements were collected from a set of residential appliances under controlled operating conditions. The acquired measurements were organized into a dataset used for feature extraction, model training, and performance evaluation.

- **Development of an Event Detection Framework**

  An event detection methodology was developed to identify appliance switching operations from measured electrical signals. The proposed approach isolates relevant state transitions and reduces the amount of data requiring subsequent processing.

- **Feature Extraction for Appliance Characterization**

  A set of electrical features was extracted from detected switching events to characterize appliance behavior. These features provide a compact representation of appliance signatures suitable for machine learning applications.

- **Development of a Random Forest Appliance Classifier**

  A supervised machine learning model based on the Random Forest algorithm was developed and trained using experimentally acquired data. The classifier was designed to identify appliances from event-based electrical signatures while maintaining low computational complexity.

- **Experimental Validation of the Proposed Framework**

  The complete framework, including the NESNAS platform, event detection module, feature extraction process, and Random Forest classifier, was experimentally evaluated using the acquired appliance dataset. The obtained results demonstrate the feasibility of the proposed approach for residential appliance identification.

1.10 Chapter Summary

In summary, this thesis demonstrates the feasibility of combining a dedicated low-cost acquisition platform with event-based NILM techniques and machine learning methods for appliance-level energy monitoring. The proposed framework contributes toward the development of practical and scalable solutions capable of supporting real-time residential energy management systems, smart home applications, and future edge-based energy monitoring architectures.

Chapter 2

Hardware Architecture and Component Selection

2.1 Introduction

The foundation of any robust Non-Intrusive Load Monitoring (NILM) system lies in the precision and reliability of its physical data acquisition layer. Before machine learning algorithms can successfully identify complex appliance footprints, the raw electrical grid signals must be captured, conditioned, and digitized with exacting fidelity. This chapter details the comprehensive hardware architecture and component selection for the NESNAS platform, transitioning the theoretical requirements of Edge-AI energy monitoring into a functional, deployed embedded system.

The discussion begins with an overview of the data acquisition architecture, followed by a detailed examination of the galvanically isolated sensing subsystem responsible for capturing phase-coherent voltage and current waveforms. It then outlines the design of a custom Analog Front-End (AFE) developed to safely condition these bipolar analog signals, and justifies the integration of a high-resolution 24-bit Analog-to-Digital Converter (ADC) to ensure micro-volt precision. Finally, the chapter addresses the local human-machine interface and the strict communication bus allocation strategies employed to guarantee uninterrupted, deterministic telemetry transfer to the Edge processor. Together, these hardware design choices form the stable physical metrology layer necessary to support the advanced digital signal processing and machine-learning classification algorithms introduced in the subsequent chapters.

2.2 System Architecture Overview

The acquisition of accurate electrical measurements is an important requirement for event-based NILM systems. The performance of machine learning algorithms, including Random Forest classifiers, depends on the quality of the measured voltage and current signals used for feature extraction. Measurement errors caused by waveform distortion, phase shifts, noise, or loss of harmonic information may affect the extracted features and consequently reduce appliance classification accuracy. For this reason, the ”NESNAS” hardware platform was designed to provide reliable signal acquisition, electrical isolation, and accurate digital conversion suitable for appliance monitoring applications.

The data acquisition architecture consists of three main stages. First, voltage and current waveforms are measured in the utility domain using non-intrusive sensing devices. The acquired analog signals are then processed by a custom analog front end AFE, where signal conditioning functions such as biasing and filtering are applied. Finally, the conditioned signals are digitized by an ADC and transferred to a Raspberry Pi through an Serial Peripheral Interface (SPI) communication interface at a sampling rate of 1600 SPS. The resulting digital measurements provide the input data used for subsequent event detection, feature extraction, and appliance classification.

2.3 Voltage and Current Measurement Subsystem

Accurate extraction of voltage (V) and current (I) features requires synchronized measurements. To achieve this, the sensing subsystem was designed with galvanic isolation while ensuring faithful signal acquisition. This design supports precise estimation of electrical parameters and provides the information necessary for event detection and appliance identification.

2.3.1 Voltage Measurement Using the ZMPT101B Sensor

The measurement of 230 V AC mains voltage requires galvanic isolation between the utility supply and the low-voltage electronics used for data acquisition and processing. Direct voltage sensing approaches based on resistive dividers can provide a scaled representation of the mains voltage but do not offer electrical isolation between the power network and the acquisition circuitry. For residential monitoring applications, an isolated sensing solution is therefore preferred to improve operational safety and protect the measurement hardware.

To address this requirement, the NESNAS platform employs the ZMPT101B active voltage transformer module [2], shown in Figure 2.1. The wiring diagram for this module is shown in Figure 2.2, where it is connected in parallel with the mains supply, enabling voltage measurement without modification of the appliance wiring. This configuration satisfies the non-intrusive monitoring principle of NILM while providing a scaled voltage signal suitable for subsequent conditioning and digitization.

The ZMPT101B provides galvanic isolation through transformer action, converting the mains voltage into a proportional low-level signal with a nominal secondary current of approximately 2 mA. As shown in the internal circuit of Figure 2.3, the transformer output is conditioned by a dual LM358 operational amplifier stage. This circuitry amplifies the measured signal and provides a low-impedance analog output suitable for subsequent signal conditioning and analog-to-digital conversion.

Accurate phase relationships between voltage and current measurements are important for the calculation of electrical features such as active power (P), reactive power (Q), and power factor. These quantities are commonly used during event detection and appliance characterization. Phase errors introduced by the sensing subsystem may affect the estimation of these features and reduce the separability of appliances with different electrical characteristics, such as predominantly resistive loads (e.g., a kettle) and loads containing inductive components (e.g., a hairdryer motor). According to the manufacturer specifications summarized in Table 2.1, the ZMPT101B exhibits a low phase error, making it suitable for power measurement applications.

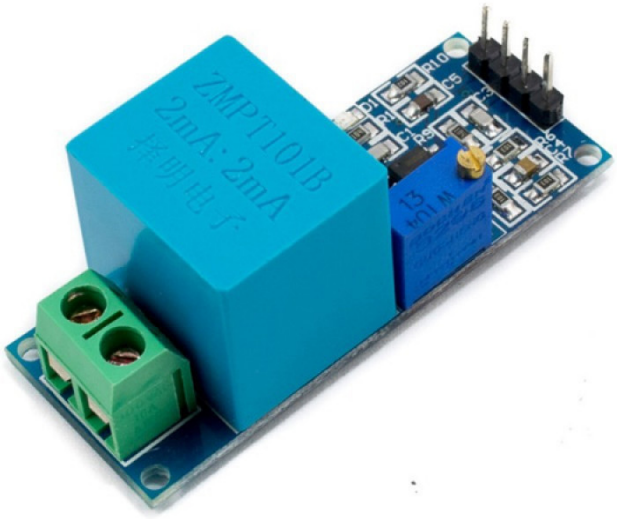


Figure 2.1: Active voltage transformer module (ZMPT101B), Image source: [8].

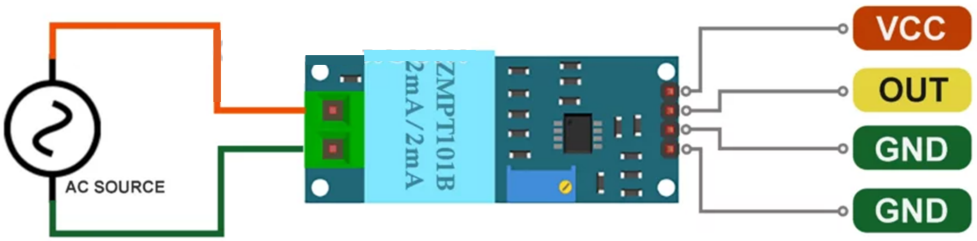


Figure 2.2: External connection diagram of the ZMPT101B module, Adapted from [8].

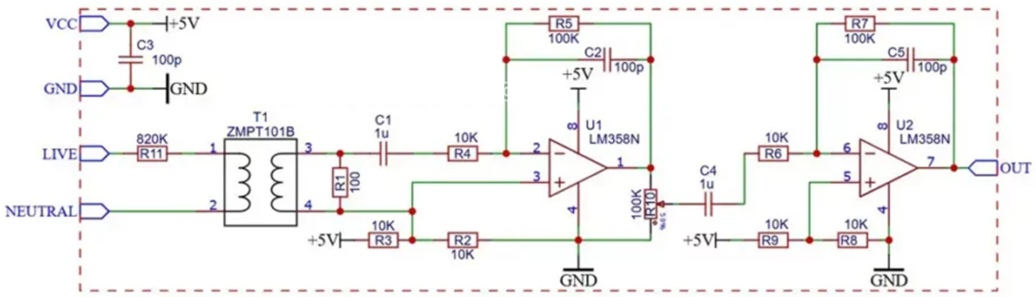


Figure 2.3: Internal schematic of the ZMPT101B active module, Source: [8].

Table 2.1: Critical Metrology Specifications of the ZMPT101B Transformer

Parameter	Rated Value	Engineering Significance
Rated Input/Output	2 mA	Provides a symmetrical 1:1 current transfer ratio.
Phase Angle Error	$\leq 20'$ (Minutes)	Ensures temporal phase coherence for robust Active/Reactive power extraction.
Linearity Range	0 ~ 1000 V	High dynamic range accommodating residential voltage sag transients.
Isolation Voltage	4000 V	Provides a substantial dielectric safety margin.

2.3.2 Current Measurement Using SCT-013 Split-Core Transformer

The NESNAS platform employs the SCT-013 split-core current transformer [27], shown in Figure 2.4, for non-intrusive current measurement. Its split-core construction enables installation around an energized conductor without direct electrical contact, allowing current measurements to be acquired without altering the existing wiring configuration.



Figure 2.4: Split-core current transformer (SCT-013-030) [27].

The SCT-013 operates as a current transformer, producing a secondary signal proportional to the current flowing through the monitored conductor. The voltage-output version of the sensor was selected to simplify the AFE design and provide a conditioned output signal within the input range of the acquisition circuitry. The resulting output is approximately 0–1 V Alternating Current (AC) and is suitable for subsequent signal conditioning and digitization.

The selection of the sensor measurement range was based on the characteristics

of the residential test environment. Although SCT-013 variants with higher current ratings are available, typical residential circuits in Algeria are protected by 16 A or 32 A circuit breakers. Consequently, the **SCT-013-030 (30 A)** model was selected to provide a measurement range consistent with the expected operating conditions. Matching the sensor range to the anticipated load currents improves measurement resolution for low-current operating states and enhances the quality of the acquired signals used for feature extraction and appliance classification.

Table 2.2: Critical Metrology Specifications of the SCT-013-030 Transformer

Parameter	**Rated Value**	**Engineering Significance**
Rated Input Current	0 ~ 30 A	Aligns with standard residential circuit breaker limits (16 A / 32 A).
Rated Output Voltage	0 ~ 1 V	Internal burden resistor stabilizes temperature drift for reliable footprinting.
Non-linearity	±3%	While a 3% error exists, the Machine Learning (ML) pipeline prioritizes relative harmonic ratios over absolute amplitude precision, ensuring cost-effective classification.
Dielectric Strength	6000 V AC/1min	Maximum insulation barrier between the live phase wire and the edge device.

The output of the SCT-013 is a bipolar AC signal and therefore requires Direct Current (DC) biasing before analog-to-digital conversion. A biasing network, shown in Figure 2.5, is used to shift the signal into the input range of the acquisition circuitry. The implementation of this biasing stage is described in Section 2.6.

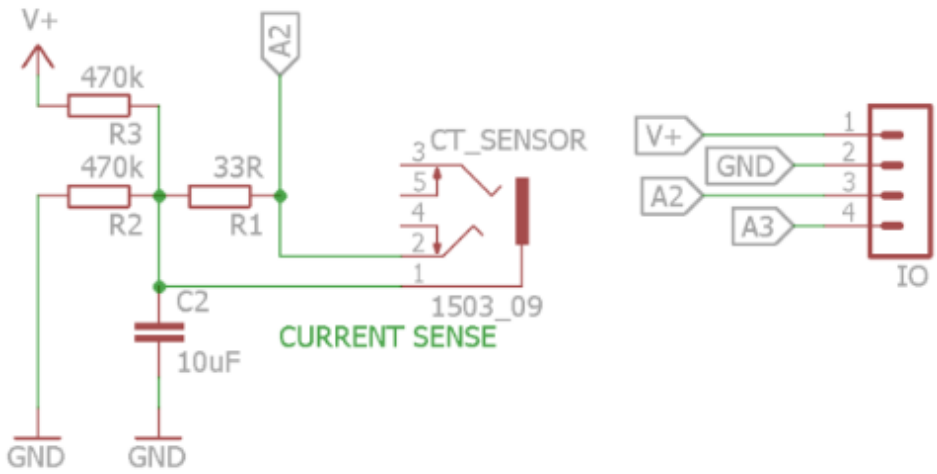


Figure 2.5: Standard DC bias conditioning circuit Schematic adapted from [18].

2.4 Analog-to-Digital Conversion Using the ADS1256

The conditioned analog signals must be converted into digital samples before further processing can be performed. Conventional microcontroller-integrated ADCs typically provide 10- or 12-bit resolution, which may be sufficient for basic power monitoring applications. For a 3.3 V input range, a 10-bit ADC provides a quantization step of approximately 3.22 mV. Consequently, waveform variations with amplitudes on the order of a few millivolts may not be represented accurately. Since event-based NILM relies on electrical features derived from waveform shape and harmonic content, a higher-resolution ADC was selected to improve measurement precision and preserve information relevant to appliance identification.

To provide high-resolution signal acquisition, the NESNAS platform employs the Texas Instruments ADS1256, a dedicated 24-bit Delta-Sigma (ΔΣ) ADC module [25], shown in Figure 2.6.

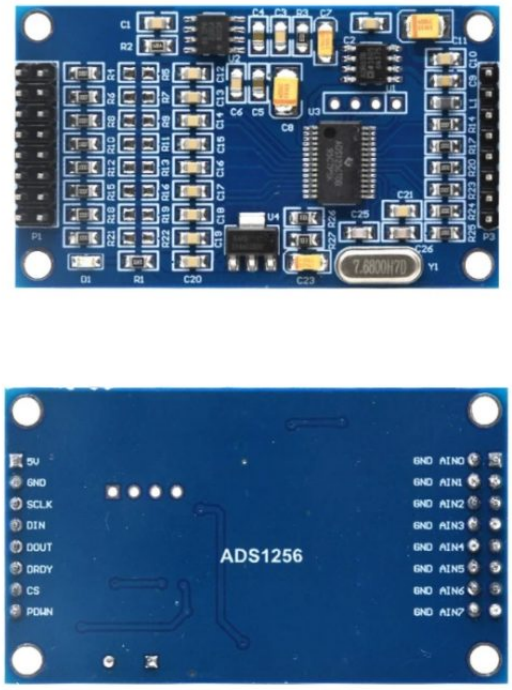


Figure 2.6: The 24-bit ADC module (ADS1256) [25].

Using a 5 V reference voltage, the ADS1256 provides a theoretical least significant bit (LSB) value of approximately 0.29 µV. Although the effective measurement resolution is limited by noise and other non-ideal effects, the high nominal resolution of the device allows small variations in the acquired voltage and current signals to be represented more accurately than with conventional microcontroller-integrated ADCs.

The ADS1256 supports sampling rates of up to 30,000 SPS. In the implemented acquisition system, measurements are acquired at 1600 SPS to satisfy the requirements of event detection and feature extraction while maintaining manageable computational and storage requirements. The higher sampling capability of the ADS1256 provides flexibility for future system development and supports accurate waveform acquisition prior to software processing.

Table 2.3: Critical Digital Conversion Specifications of the ADS1256

Parameter	Rated Value	Engineering Significance
Resolution	24-bit ($\Delta\Sigma$)	Provides 16.7×10^6 steps, capturing micro-volt level state transients.
Max Sample Rate	30,000 SPS	Hardware oversampling supports the stabilized 1,600 SPS software pipeline.
Communication	SPI	High-speed synchronous serial bus capable of sustaining continuous telemetry.
Programmable Gain	Up to 64x	Allows dynamic analog amplification of weak signatures prior to inference.

2.5 The Human-Machine Interface and Bus Allocation

The NESNAS platform integrates a local human–machine interface Human-Machine Interface (HMI) to provide operating status information and user interaction capabilities. Because the data acquisition subsystem relies on continuous communication with the ADS1256 ADC, the selection of display hardware must consider potential resource conflicts that could affect acquisition performance.

2.5.1 Display Selection and Bus Contention

The primary design constraint is preserving uninterrupted SPI communication between the Raspberry Pi and the ADS1256. An SPI-based LCD would require frequent switching of Chip Select (CS) lines and additional bus transactions for display updates, thereby competing with the data acquisition process for access to the shared interface. This contention could introduce timing jitter into the 1,600 SPS sampling stream, degrading the temporal consistency of the acquired measurements and potentially affecting the performance of the event-detection and appliance-identification algorithms.

A 2.42-inch OLED display operating through the I^2C interface was selected for the human–machine interface (HMI) [15], as shown in Figures 2.7 and 2.8. The use of a dedicated I^2C communication bus separates display operations from the SPI interface used by the ADS1256 ADC. This configuration reduces communication contention between the display and data acquisition subsystems and allows the SPI bus to remain dedicated to the transfer of measurement data.

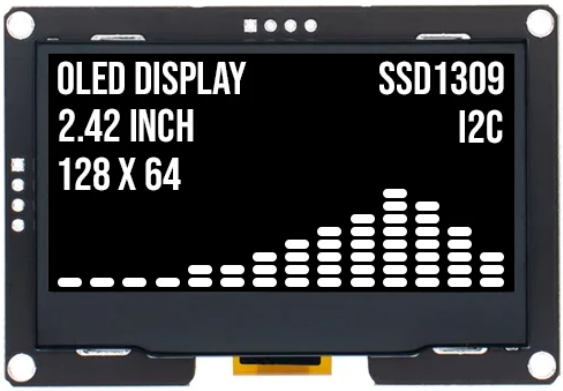


Figure 2.7: Front view of the 2.42-inch OLED [23].

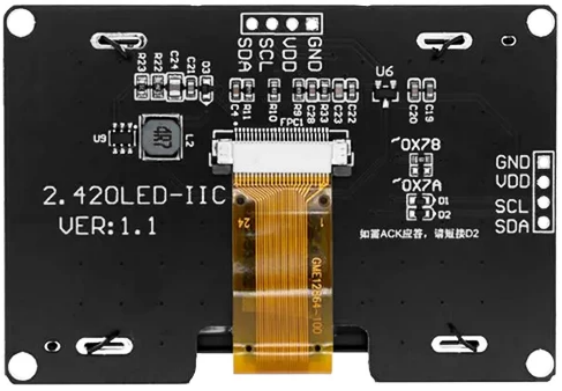


Figure 2.8: Rear view detailing the isolated I^2C interface [23].

2.5.2 Tactile Navigation Array

The HMI also integrates a four-button interface that enables user interaction with the system, including event annotation and control of data acquisition procedures. Due to component availability constraints, individual push-button modules (Figure 2.10) were used instead of the integrated button array shown in Figure 2.9.

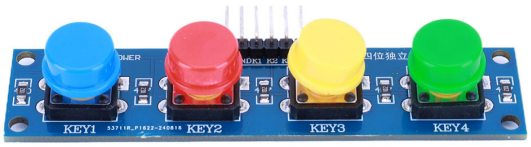


Figure 2.9: Ideal monolithic 4-button array for interface navigation [17].

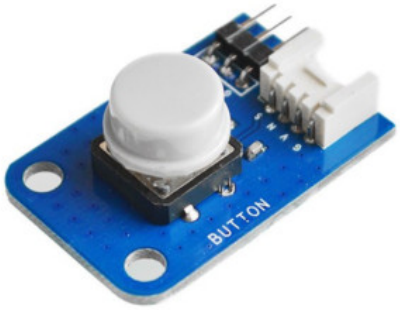


Figure 2.10: Implemented single-button logic module utilized via hardware interrupts [17].

The push-button modules are connected directly to the Raspberry Pi General-Purpose Input/Output (GPIO) pins, eliminating the need for additional interface controllers. Internal pull-up resistors are used, and button events are detected through GPIO interrupt mechanisms.

Mechanical contact bounce can generate multiple interrupt events during a single button press. To mitigate this effect, a software debouncing interval of 200 ms was

implemented within the Python event-handling routines. This approach ensures reliable button operation and prevents unintended multiple detections while maintaining responsiveness of the data acquisition and classification processes.

2.6 Analog Front-End Design

The voltage and current sensors produce bipolar AC signals, whereas the ADS1256 ADC operates within a unipolar input range. Consequently, signal conditioning is required before digitization to ensure compatibility between the sensor outputs and the acquisition circuitry. In particular, negative voltage excursions may exceed the permissible ADC input range and adversely affect measurement accuracy or device operation.

To address these requirements, a custom AFE was designed and implemented, as shown in Figure 2.11. The AFE performs the necessary signal conditioning functions prior to analog-to-digital conversion.

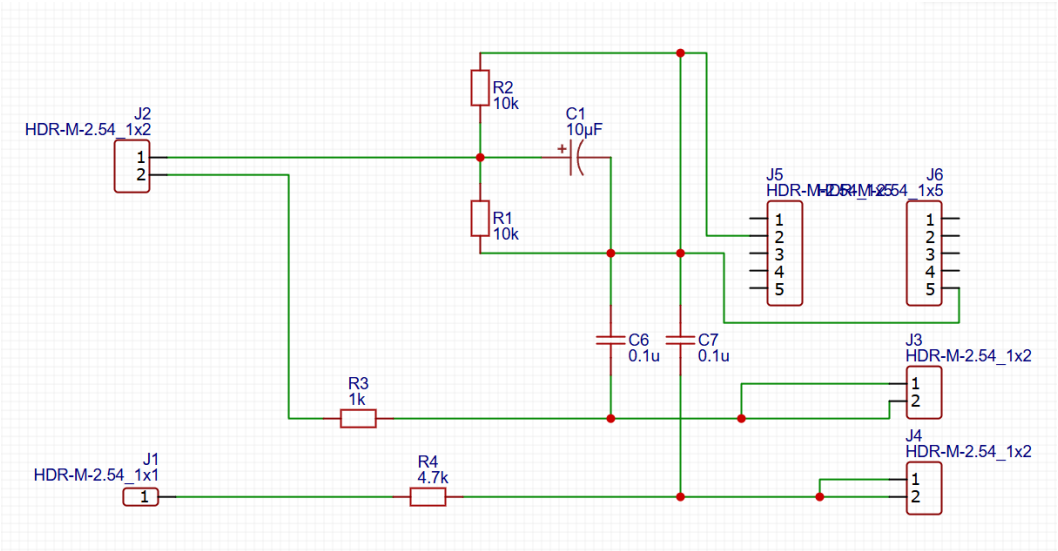


Figure 2.11: Schematic of the custom AFE.

In addition to protecting the ADC input stage, the AFE performs signal conditioning through DC biasing. A 1.65 V offset is generated using a matched 10 kΩ resistor divider network, positioning the sensor outputs at the midpoint of the ADC operating range. This biasing scheme enables bipolar AC waveforms to be sampled by the unipolar ADC while preserving their amplitude and phase characteristics.

The bias voltage is calculated as

$$V_{bias} = V_{cc} \times \frac{R_1}{R_1 + R_2} = 3.3\ \text{V} \times \frac{10\ \text{k}\Omega}{20\ \text{k}\Omega} = 1.65\ \text{V} \tag{2.1}$$

The AFE also attenuates high-frequency components before digitization. According to the Nyquist–Shannon sampling theorem [22], the maximum frequency that can be represented without aliasing is half the sampling frequency. For the implemented acquisition rate of 1600 SPS, this corresponds to a Nyquist frequency of 800 Hz.

To reduce high-frequency noise, the AFE incorporates an RC low-pass filter with a cutoff frequency of approximately 1.59 kHz. Although this cutoff frequency exceeds

the Nyquist frequency associated with the final output data rate, the ADS1256 employs a Delta–Sigma conversion architecture that internally oversamples the input signal at significantly higher rates. The converter also includes a digital sinc filter that performs decimation and anti-alias filtering before generating the output data stream.

As a result, the analog RC filter provides initial attenuation of high-frequency noise and interference, while the internal filtering stages of the ADS1256 perform the primary anti-aliasing and downsampling operations before transmitting data at 1600 SPS through the SPI interface.

$$f_c = \frac{1}{2\pi \cdot R_3 \cdot C_6} \approx 1591 \text{ Hz} \tag{2.2}$$

The selected cutoff frequency of approximately 1.59 kHz allows the fundamental 50 Hz component and its lower-order harmonics to pass with limited attenuation while reducing higher-frequency noise components. In particular, harmonics up to approximately the 20th order (1000 Hz) remain within the passband of the analog filter and may contribute useful information for appliance characterization and classification.

The PCB layout, illustrated in Figure 2.12, was designed to support reliable signal acquisition and conditioning. The fabricated prototypes shown in Figures 2.13 and 2.14 implement the proposed analog front-end circuitry and data acquisition interfaces while following standard layout practices intended to reduce noise and maintain signal integrity.

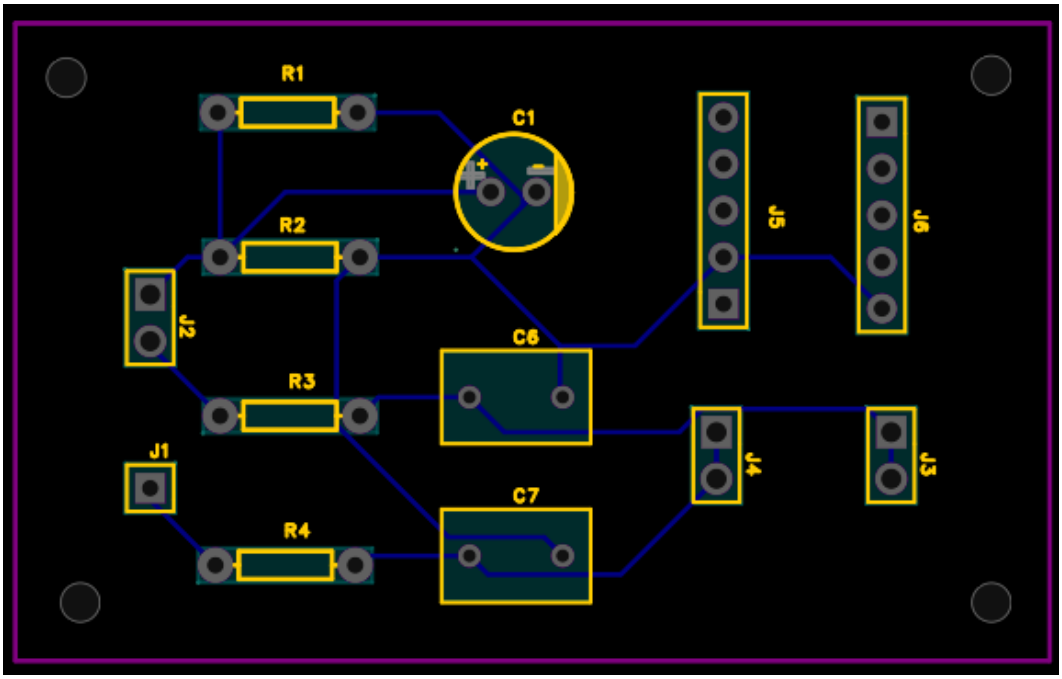


Figure 2.12: GERBER layout of the AFE designed in EasyEDA.

Prior to the fabrication of the final printed circuit board, the analog conditioning circuits and digital routing pathways were physically validated using a hand-soldered prototype. As shown in Figures 2.15 and 2.16, this initial physical realization established the baseline wiring topology. It successfully segregated the 3.3 V and ground distribution rails for the local interface while isolating the critical analog pathways connecting the ZMPT101B and SCT-013 transducers to the ADS1256 ADC. Testing this

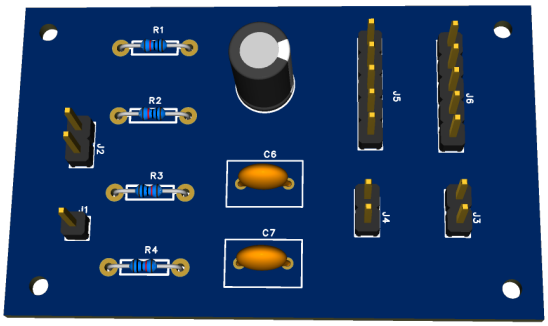


Figure 2.13: 3D render of the AFE component side.

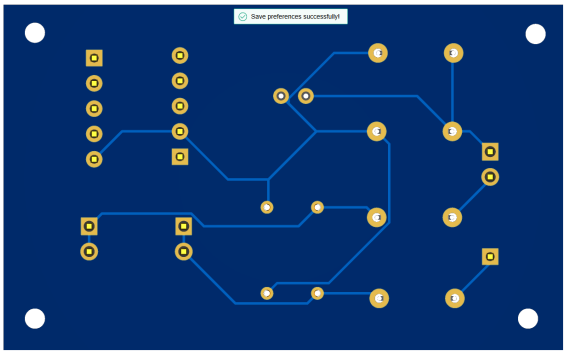


Figure 2.14: 3D render of the AFE solder side.

physical iteration confirmed the viability of the DC-biasing network and anti-aliasing filters, serving as the functional foundation for the final PCB layout.

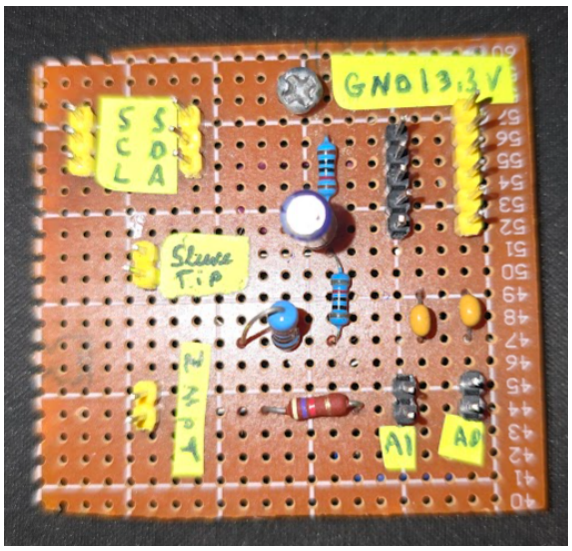


Figure 2.15: Physical realization of the AFE prototype on perfboard: front view showing component placement.

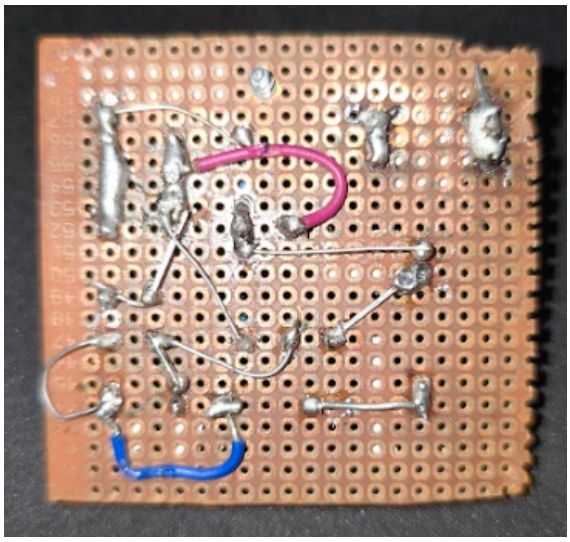


Figure 2.16: Physical realization of the AFE prototype on perfboard: back view showing hand-soldered routing.

2.7 Chapter Summary

This chapter detailed the foundational hardware engineering required to support an autonomous Edge-AI NILM instrument. By systematically selecting components capable of high-fidelity signal acquisition—such as the ZMPT101B and SCT-013 sensors paired with a 24-bit ADS1256 ADC—the metrology pipeline ensures that highly discriminative harmonic features are preserved. Additionally, custom AFE conditioning and strict bus allocation protocols were implemented to safeguard the data stream from aliasing and interface-induced jitter. With the physical sensing layer secured, the subsequent chapter will outline the embedded software integration and continuous signal processing framework that processes these raw analog inputs.

Chapter 3

NESNAS Realization

3.1 Introduction

With the physical metrology hardware and analog front-end architecture established, the focus of system development shifts toward the physical realization and digital integration of the instrument. This chapter details the comprehensive prototyping of the NESNAS platform, documenting its evolution from discrete electronic components into a fully autonomous, Edge-AI computational instrument.

The chapter begins by specifying the embedded computing architecture, utilizing a Raspberry Pi 4 Model B configured with a customized, headless Linux environment designed to maximize processing resources. It then chronicles the iterative mechanical design process, illustrating how initial CAD concepts were refined to resolve issues regarding component strain, high-voltage isolation, and active thermal management. Following the mechanical assembly, the chapter details the critical electrical integration pathways, focusing on sensor connections, interface pin allocation, and strict digital bus segregation.

Finally, the discussion transitions to the embedded software stack. A sophisticated, multi-threaded architecture is presented, designed specifically to circumvent the Python Global Interpreter Lock (GIL). By decoupling the deterministic 1,800 SPS data acquisition loop from the slower User Interface and Machine Learning inference routines, the system maintains phase-coherent feature extraction . The chapter concludes by demonstrating the platform's local Human-Machine Interface (HMI), proving the system's capability to execute autonomous dataset generation and real-time Non-Intrusive Load Monitoring entirely at the network edge.

3.2 Embedded Computing Platform

3.2.1 Raspberry Pi Hardware

The Raspberry Pi 4 Model B was selected because it provides a balance between computational performance, software ecosystem maturity, GPIO/SPI availability, and low deployment cost. The quad-core Cortex-A72 processor offers sufficient computational resources for real-time signal processing and Random Forest inference while maintaining significantly lower power consumption and cost. The main specifications of the Raspberry Pi 4 Model B are summarized in Table 3.1.

Table 3.1: Key Specifications of the Raspberry Pi 4 Model B

Parameter	Specification
Processor	Broadcom BCM2711, Quad-core ARM Cortex-A72 (1.5 GHz)
Memory	4 GB LPDDR4 SDRAM
Wireless Connectivity	Dual-band Wi-Fi (802.11ac), Bluetooth 5.0 BLE
Ethernet	Gigabit Ethernet
USB Ports	2 × USB 3.0, 2 × USB 2.0
GPIO Header	40-pin GPIO
Communication Interfaces	SPI, I^2C, UART
Storage	microSD card
Power Supply	5 V DC via USB-C
Dimensions	85.6 mm × 56.5 mm

3.2.2 Nesnas Operating System

The embedded platform operates using Raspberry Pi OS Lite, a lightweight Linux environment configured for headless operation. The operating system was deployed on a 64 GB Class 10 UHS-I microSD card using the official Raspberry Pi Imager toolchain, with SSH access and network parameters preconfigured during installation. The UHS-I storage medium provides adequate bandwidth for local logging and temporary buffering of acquired measurements.

To maximize the computational resources available for data acquisition, signal processing, and machine learning inference, the graphical desktop environment and non-essential background services were disabled.

A dedicated Python virtual environment (`nesnas_env`) was created to isolate all software dependencies required by the platform, including `spidev` for SPI communication, `numpy` for numerical processing, and `scikit-learn` for the Random Forest classifier. This isolated software environment improves reproducibility and simplifies dependency management during future software updates and experimental validation.

Collectively, the customized operating system, software environment, and supporting services developed for the embedded platform are hereafter referred to as Nesnas OS.

3.2.3 Remote System Administration

Because the Edge-AI prototype operates without physical input peripherals, a reliable remote administration mechanism was required. Tailscale, a zero-configuration mesh Virtual Private Network (VPN) built on the WireGuard cryptographic protocol [5], was integrated directly into the core deployment.

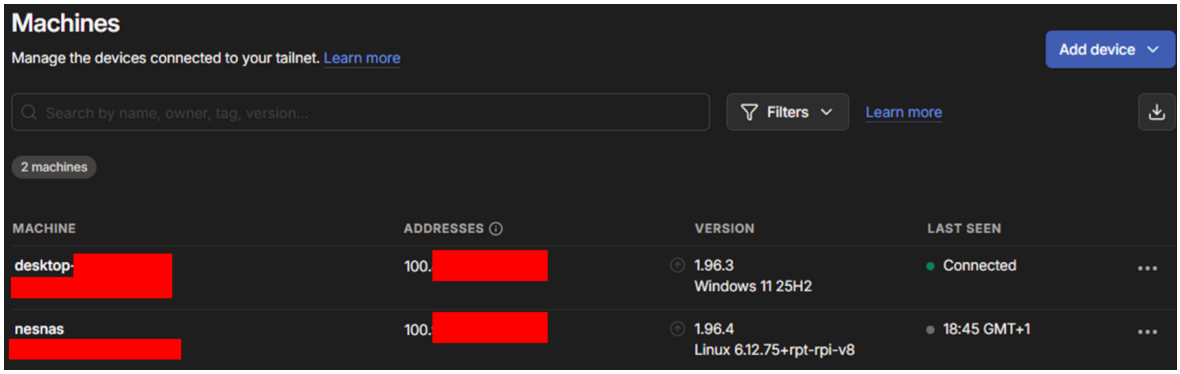


Figure 3.1: The Tailscale administrative dashboard, displaying the active connection to the `nesnas` Edge node.

As illustrated in Figure 3.1, this architecture enables secure connectivity across different network environments, including those protected by network address translation (NAT) and firewall restrictions. By authenticating onto a cryptographically secured subnet, Tailscale assigns a static `100.x.x.x` IP address. This enables secure SSH access, allowing the deployed machine learning pipeline to be monitored, updated, and debugged regardless of the physical deployment network.

To continuously refine the feature extraction algorithms directly on the Nesnas, a remote development pipeline was established. Utilizing Visual Studio Code (VS Code) with the Remote-SSH extension, the remote file system of the Raspberry Pi was mounted to a localized workstation. This granted access to syntax highlighting and version control without burdening the Pi's Central Processing Unit (CPU) with rendering an Integrated Development Environment (IDE) graphical interface.

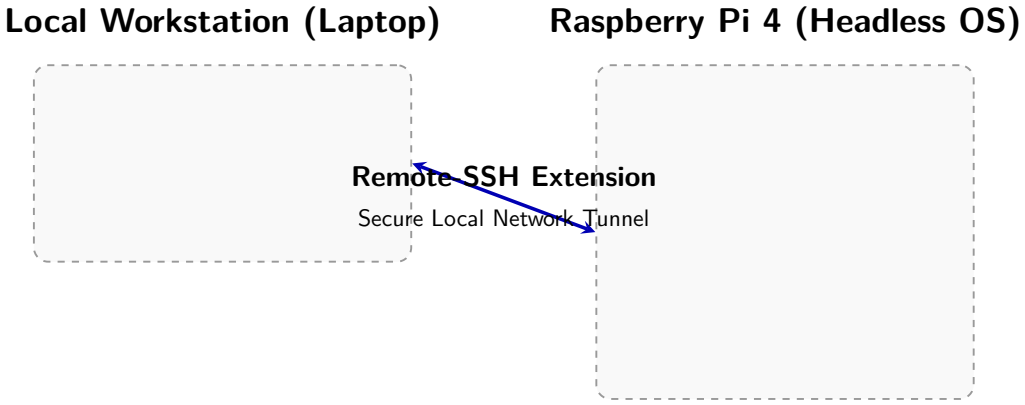


Figure 3.2: Architectural representation of the remote development environment.

3.3 Mechanical Design and Physical Realization

The mechanical structure of the NESNAS instrument was developed to ensure physical protection, secure mounting of high-voltage electronics, effective cable routing, and reliable thermal management for the Edge-AI processor. The enclosure design underwent multiple iterations using Computer-Aided Design (CAD) to resolve issues related to mechanical stress on analog conditioning circuits and to optimize the internal topology (a detailed breakdown of the CAD iterations and enclosure evolution is

provided in Appendix A).

The finalized mechanical architecture utilizes a modular, bottom-mounted layout where the principal electronic components are securely anchored to the enclosure base. This topology prevents strain on the high-voltage wiring during maintenance and ensures stable operation of the Analog Front-End (AFE).

To maintain stable operating temperatures during continuous data acquisition and Random Forest inference, active thermal management was integrated into the enclosure. A 30×30 mm cooling fan was mounted directly above the Raspberry Pi Cortex-A72 processor, ensuring optimal airflow and preventing thermal throttling, as detailed in Table 3.2.

Table 3.2: Thermal Performance of the Raspberry Pi 4 Under Continuous Inference Load

Cooling Configuration	**Average Core Temp (°C)**	**Inference Stability**
Passive Heatsinks Only	51.0	Possible Clock Jitter / Dropped Frames
Active 30 mm Elevated Fan	41.0	Optimal / Low Latency

The physical realization of the instrument features dedicated external interface panels. The front panel houses the OLED display and the four-button tactile array for local Human-Machine Interface (HMI) interaction. The side panels provide structured cable management, featuring dedicated access ports and integrated spools for the ZMPT101B and SCT-013 transducer leads. A CNC-machined transparent acrylic window was also installed to allow visual diagnosis of internal status LEDs without breaching the enclosure.

The finalized, fabricated NESNAS prototype is presented in Figures 3.3 and 3.4. The resulting system successfully integrates the embedded computing platform, sensing interfaces, and thermal ventilation strategy into a compact, portable instrument suitable for real-world deployment.

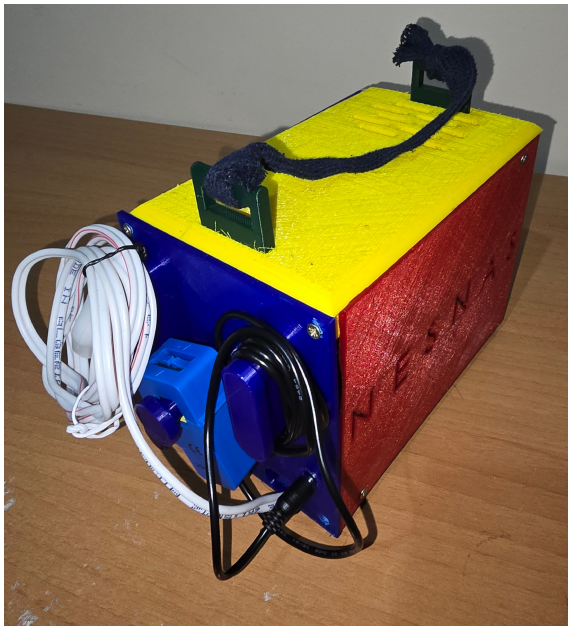


Figure 3.3: Front and top view of the fabricated NESNAS instrument, featuring the local HMI and ventilation.

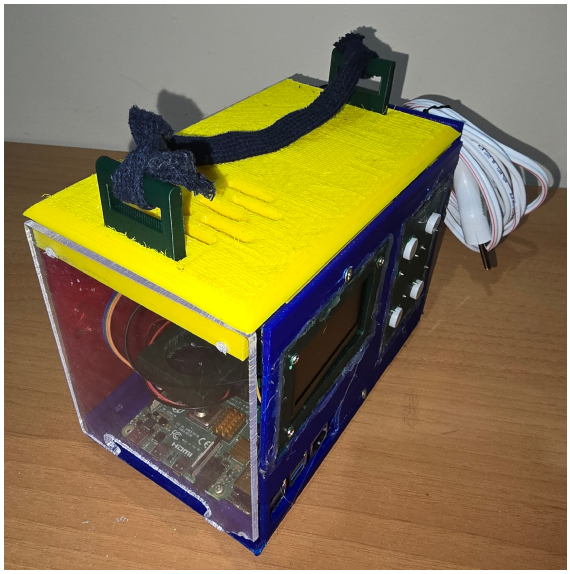


Figure 3.4: Rear view demonstrating sensor integration, panel-mount audio jacks, and cable management.

3.3.1 Current Sensor Interface and Connector Mapping

To provide a modular and serviceable interface for the SCT-013 current transformer, a panel-mounted 3.5 mm audio jack was integrated into the enclosure. Because commercially available panel-mount audio connectors do not universally follow a standardized rear-terminal arrangement, the connector terminals were experimentally verified prior to integration.

Terminal identification was performed using continuity measurements to establish the correspondence between the internal Tip and Sleeve contacts and the external solder lugs. By probing the internal contact surfaces and monitoring continuity at the rear terminals, the correct electrical mapping of the connector was determined. This verification step ensured that the current transformer output could be connected to the Analog Front End (AFE) without polarity ambiguity or wiring errors.

Following terminal identification, the connector was permanently integrated into the sensing subsystem. Jumper leads were soldered to the verified Tip and Sleeve terminals and routed directly to the AFE. The Sleeve terminal was connected to the 1.65 V bias reference network, while the Tip terminal was connected to the conditioned signal path leading to the acquisition circuitry.

The external interface was completed by inserting the SCT-013 current transformer plug into the panel-mounted connector. This arrangement provides a clean and modular connection between the external sensing hardware and the internal measurement circuitry while allowing rapid sensor replacement and maintenance without requiring access to the enclosure interior. The resulting design improves system serviceability and supports straightforward deployment of the current sensing subsystem.

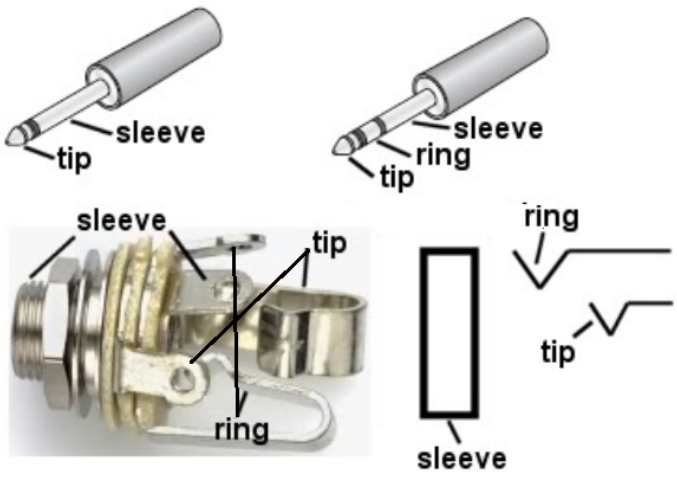


Figure 3.5: Anatomy of the 3.5 mm audio jack used to anchor current telemetry to the custom AFE. [7].

3.3.2 ADC Integration and Analog Pathways

The ADS1256 served as the primary data acquisition device within the NESNAS platform, providing high-resolution digitization of the conditioned voltage and current waveforms. An important electrical design decision was the adoption of a single-ended acquisition topology. The custom Analog Front End (AFE) applies a 1.65 V DC offset to the bipolar sensor outputs, translating them into the unipolar input range accepted by the ADC.

To establish a common measurement reference, channel `AIN7` was connected directly to the Raspberry Pi system ground. The conditioned voltage signal from the ZMPT101B was routed to `AIN0`, while the conditioned current signal from the SCT transformer interface was connected to `AIN1`. Consequently, both channels were sampled relative to `AIN7`, enabling independent acquisition without requiring differential measurement circuitry. Following digitization, this DC component is removed in software to recover the original AC waveforms for feature extraction. The selected architecture provides a practical balance between hardware simplicity, measurement accuracy, and computational efficiency.

The complete analog signal pathways are summarized in Table 3.3.

Table 3.3: Analog Metrology Pathways (Sensors to AFE to ADS1256)

Source	Destination	Pin Mapping	Engineering Note
Raspberry Pi Ground	ADS1256	`AIN7`	Common reference channel for single-ended measurements.
ZMPT101B (Analog Output)	AFE → ADS1256	`AIN0`	Conditioned voltage waveform sampled relative to `AIN7`.
SCT Current Sensor	AFE → ADS1256	`AIN1`	Conditioned current waveform sampled relative to `AIN7`.
Sensor Return Path	AFE Ground Rail	GND	Connected to the common ground reference shared with `AIN7`.

3.3.3 Digital Communication Interfaces

Digital communication within the NESNAS platform utilizes two primary buses—SPI and Inter-Integrated Circuit (I2C)—alongside dedicated GPIO interfaces. To preserve data integrity, careful attention was given to signal routing, minimizing electromagnetic coupling between the high-speed digital lines and the sensitive analog circuitry.

Communication between the Raspberry Pi and the ADS1256 was established via the hardware SPI bus, which was routed flush against the bottom base plate. The interface utilizes Physical Pin 19 (MOSI), Pin 21 (MISO), Pin 22 (Chip Select), and Pin 23 (SCLK). A dedicated Data Ready (DRDY) interrupt line was connected to Physical Pin 11, providing hardware-level synchronization to ensure the acquisition software retrieves samples only after a completed conversion cycle.

The Organic Light-Emitting Diode (OLED) display operates on the slower I2C bus (SDA on Pin 5, SCL on Pin 3). To simplify internal wiring and reduce cable congestion, these signals were routed through dedicated transit rails on the AFE perfboard (Figure 3.6). This neatly segregates the visual output traffic from the critical SPI acquisition pathways.

User interaction is handled by four tactile push-buttons connected to Raspberry Pi GPIO Physical Pins 29, 31, 33, and 35. A centralized ground rail on a separate button perfboard distributes a common return path for these inputs. The complete digital pin allocation is summarized in Table 3.4.

Table 3.4: Digital Communication (SPI, I2C & HMI Buttons)

Signal / Protocol	**Physical Pin**	**Destination**	**Engineering Note**
I2C SCL	Pin 3	AFE Rail → OLED SCL	Routed through the AFE perfboard for front panel reachability.
I2C SDA	Pin 5	AFE Rail → OLED SDA	Routed alongside the SCL rail for display data transmission.
DRDY	Pin 11	ADS1256	Critical interrupt pin polled by the Operating System (OS) daemon thread.
SPI MOSI	Pin 19	ADS1256	Data In (Master Out Slave In).
SPI MISO	Pin 21	ADS1256	Data Out (Master In Slave Out).
CS	Pin 22	ADS1256	Chip Select; triggers the ADC for active polling.
SPI SCLK	Pin 23	ADS1256	Serial Clock maintaining synchronization.
Button 1 Signal	Pin 29	Front Panel Button	Plug-and-play OS navigation input.
Button 2 Signal	Pin 31	Front Panel Button	Plug-and-play OS navigation input.
Button 3 Signal	Pin 33	Front Panel Button	Plug-and-play OS navigation input.
Button 4 Signal	Pin 35	Front Panel Button	Plug-and-play OS navigation input.

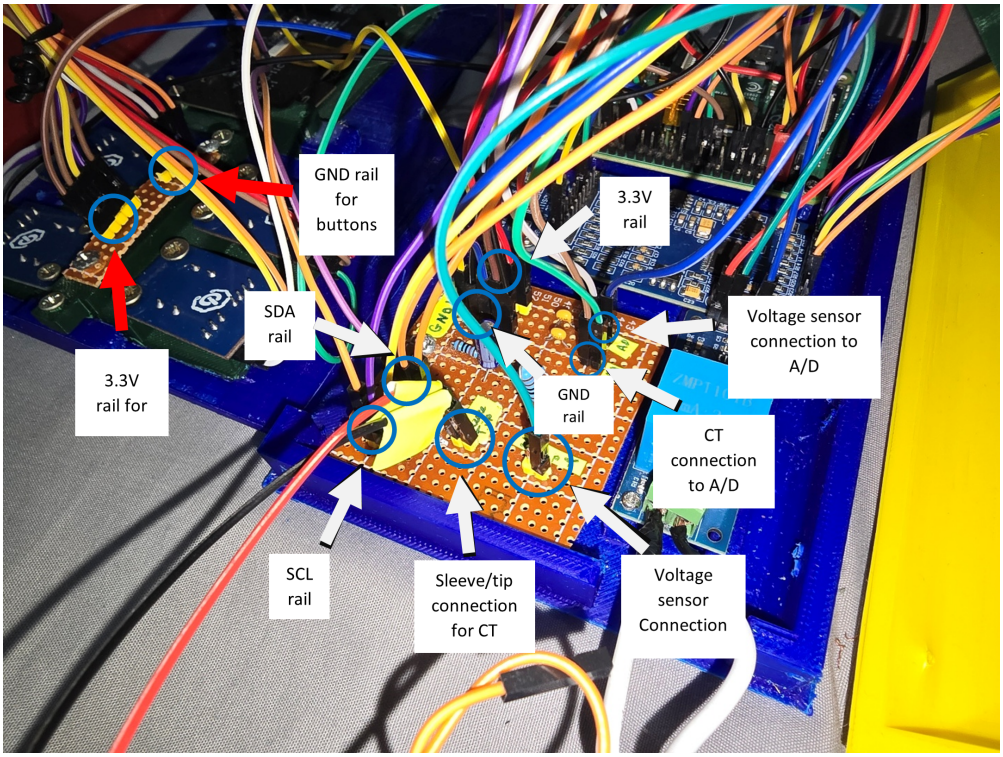


Figure 3.6: Routing topology of the AFE perfboard.

3.3.4 Power Distribution and Grounding

Reliable operation required careful power distribution and grounding to minimize electrical noise. The 5 V supply was partitioned according to functional requirements: Pin 2 was dedicated to powering the ADS1256 ADC, while Pin 4 supplied the 30×30 mm cooling fan. Pin 6 provided the fan's return path, isolating motor switching noise from the measurement circuitry.

The 3.3 V network was implemented separately. Pin 1 supplied the primary 3.3 V rail on the AFE perfboard, powering the op-amp stages, the ZMPT101B logic section, and the OLED. To accommodate space constraints, the tactile buttons were mounted on a dedicated perfboard supplied by Pin 17, with Pin 34 acting as its ground return.

Although all Raspberry Pi ground pins share a common internal ground plane, separating return paths at the wiring level heavily reduces shared ground-current coupling. On the AFE, Pin 9 established the primary ground rail for the analog sensors and the `AIN7` reference, while Pin 14 provided an isolated ground path specifically for the ADS1256. Table 3.5 summarizes this topology.

Table 3.5: Power Distribution and Ground Rails (Perfboard Routing)

Source (Raspberry Pi)	Physical Pin	Destination / Perfboard	Engineering Note
3.3 V Rail	Pin 1	AFE 3.3 V Main Rail	Powers the AFE op-amps, ZMPT logic, and OLED VCC.
5 V Rail	Pin 2	ADS1256 Power	Delivers 5 V directly to the high-precision ADC.
5 V Rail	Pin 4	30x30 mm Fan (+)	Dedicated power for the thermal convection fan.
GND	Pin 6	30x30 mm Fan (-)	Dedicated ground return for the fan to isolate motor noise.
GND	Pin 9	AFE GND Main Rail	Serves the AFE, ZMPT, and `AIN7` common reference.
GND	Pin 14	ADS1256 Ground	Isolated ground path specifically for the ADC.
3.3 V Rail	Pin 17	Button Perfboard (3.3 V)	Dedicated 3.3 V distribution rail for the four tactile buttons.
GND	Pin 34	Button Perfboard (GND)	Dedicated ground rail tying all four tactile buttons together.
GND	Pin 39	OLED Screen Ground	Direct ground return for the front panel display.

3.3.5 System Wiring and GPIO Allocation

igure 3.7 summarizes the final hardware allocation across the Raspberry Pi 4 header. This structural assignment heavily minimizes bus contention, isolates high-speed acquisition interrupts from secondary user interfaces, and maintains strict pin availability for future expansion peripherals.

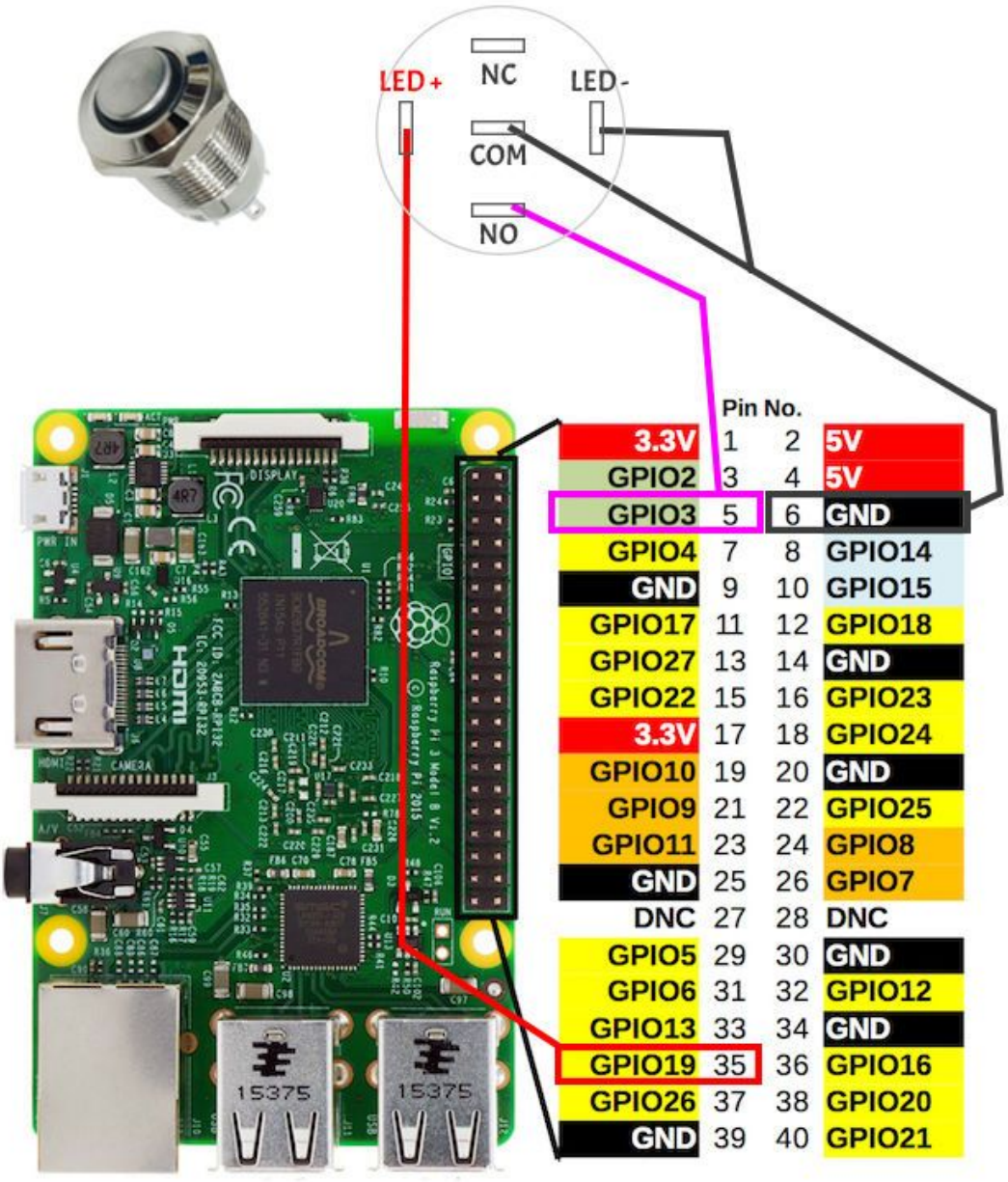
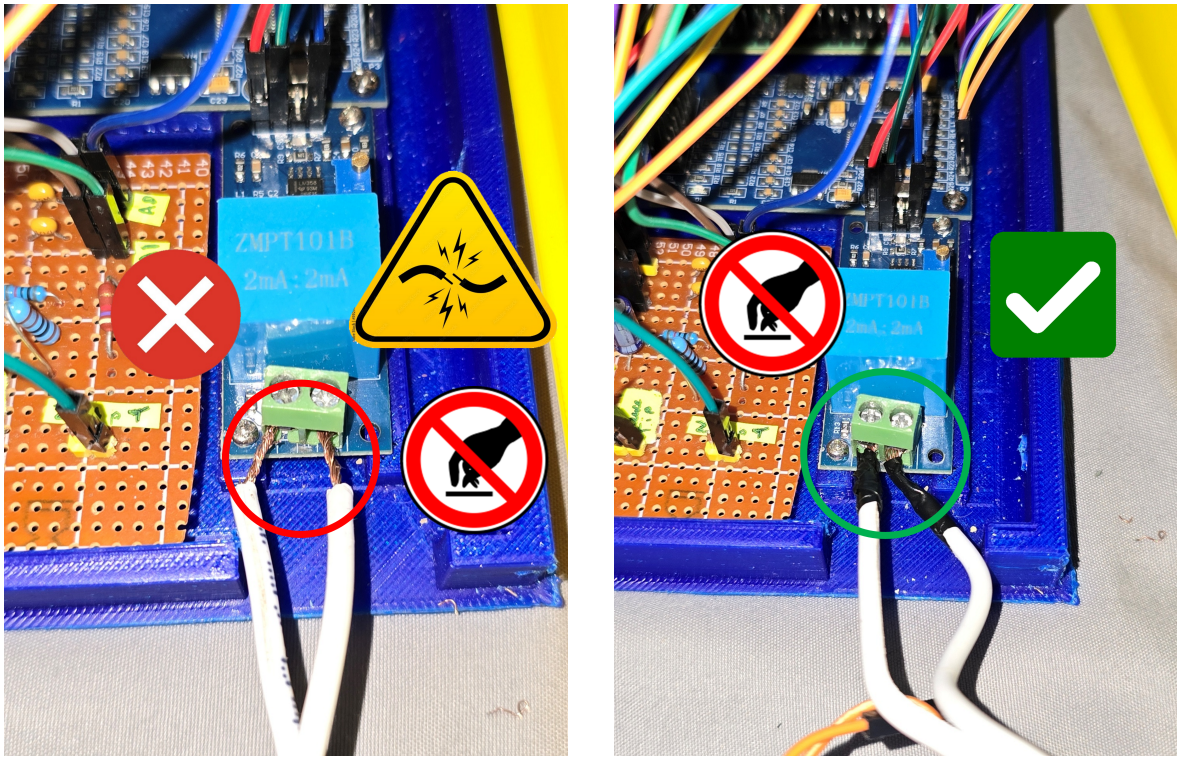


Figure 3.7: Physical pin mapping of the Raspberry Pi 4 GPIO header used in the NESNAS instrument. [11].

3.3.6 Sensor Integration and Mains Isolation

The sensing subsystem was integrated using two complementary transducers: the SCT-013 current transformer for current measurement and the ZMPT101B voltage transformer for voltage measurement. Together, these sensors provide the electrical signals required for subsequent conditioning, digitization, and NILM analysis.



Incomplete conductor termination. *Proper conductor termination.*

Figure 3.8: Comparison of conductor termination practices at the ZMPT101B mains-voltage sensing interface.

Current measurements were obtained using the SCT-013 current transformer. The sensor output was interfaced to the AFE through a modified 3.5 mm receptacle, with the signal conductor routed to the conditioning circuitry and the return path connected to the common ground rail. Because current transformers operate by measuring the magnetic field surrounding a conductor, only the live phase conductor was routed through the SCT-013 core. Passing both the live and neutral conductors through the transformer would result in magnetic field cancellation and prevent accurate current measurement.

Voltage measurements were acquired using the ZMPT101B voltage transformer module. The primary winding was connected directly across the live and neutral conductors of the 230 V AC mains supply, while the isolated low-voltage secondary output was routed to the AFE for signal conditioning. This configuration provided galvanic isolation between the mains network and the low-voltage acquisition electronics, allowing continuous voltage monitoring while reducing electrical hazards to the embedded system.

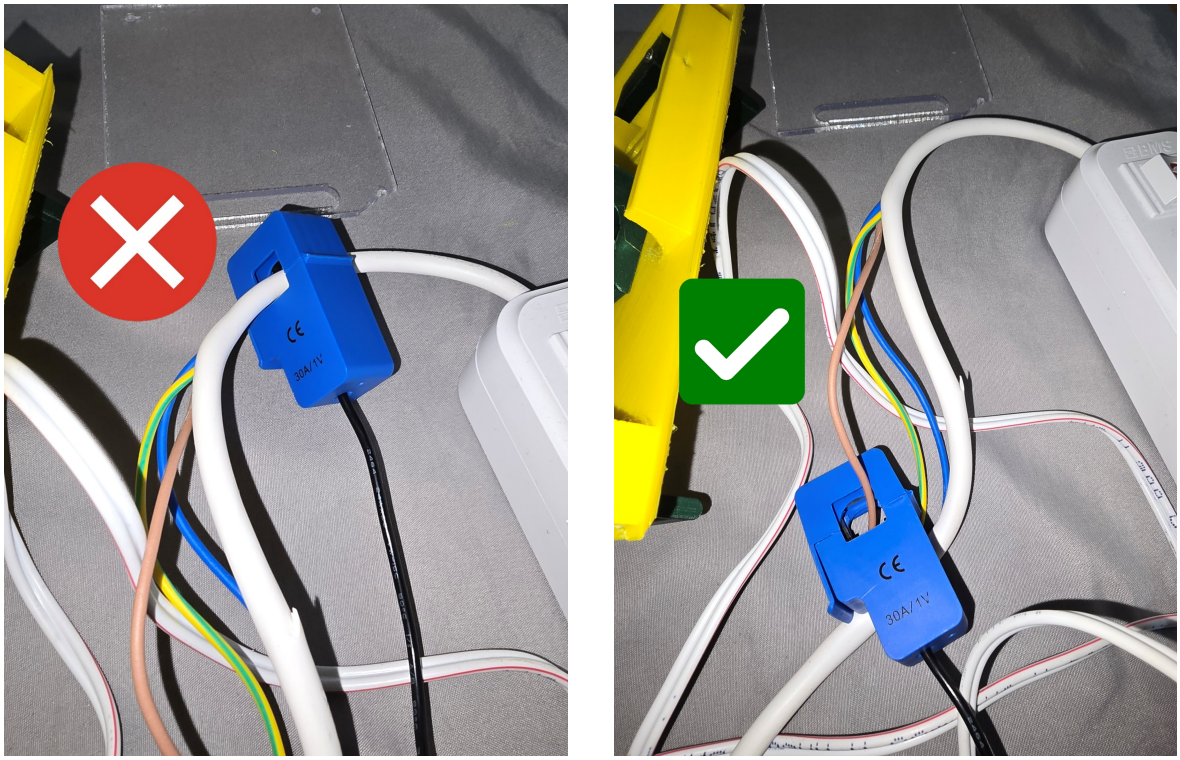
To improve operational safety and installation reliability, all mains-voltage conductors were mechanically secured and routed separately from the low-voltage electronics. Particular attention was given to conductor termination at the ZMPT101B input terminal. As shown in Figure 3.8, proper insertion and insulation of the conductors were maintained to minimize the risk of accidental contact with energized conductors and to ensure reliable long-term operation. Following isolation by the sensing transformers, the resulting low-voltage analog signals were routed to the AFE for conditioning before being digitized by the ADS1256 ADC.

The high-voltage integration pathways used by the sensing subsystem are summarized in Table 3.6.

Table 3.6: High-Voltage Mains Integration Pathways

Source	Destination	Connection	Engineering Note
Live Conductor (230 V AC)	ZMPT101B	L Input	Primary voltage sensing connection.
Neutral Conductor	ZMPT101B	N Input	Completes the voltage sensing circuit.
Live Conductor Only	SCT-013 Current Transformer	Magnetic Core	Required to prevent magnetic field cancellation and enable current measurement.

Capturing the magnetic current footprint required avoiding magnetic phase cancellation. If the SCT-013 clamp surrounds an intact residential cable, the alternating current in the live and neutral wires generate opposing magnetic fields, rendering a net flux of near zero (Figure 3.9, Left). The multisocket's PVC jacket was stripped, allowing the sensor to clamp exclusively around the live phase conductor (Figure 3.9, Right). This ensures the ADC captures the uncorrupted harmonic envelope.



Incorrect: Clamping entire cable. *Correct: Clamping isolated phase wire.*

Figure 3.9: Installation protocol for the SCT-013 Current Transformer.

3.4 Software Architecture

The Nesnas software stack was developed to transform raw electrical measurements into structured telemetry suitable for event-based appliance identification. Running on the Raspberry Pi platform, the software integrates data acquisition, digital

signal processing, telemetry management, visualization, and preparation of feature vectors for subsequent machine-learning inference within a concurrent execution framework. To satisfy the timing requirements imposed by high-rate ADC sampling, the architecture separates time-critical measurement tasks from communication and user-interface services through asynchronous processing layers. This design ensures that feature extraction, visualization, storage, downstream inference services can execute concurrently without compromising sampling integrity or system responsiveness.

The implementation was organized into four principal modules:

- **`nesnas_core.py`:** Performs hardware acquisition and digital signal processing.
- **`nesnas_os.py`:** Coordinates telemetry routing and system services.
- **`nesnas_ui.py`:** Manages local interaction through the OLED display and GPIO buttons.
- **`nesnas_web.py`:** Provides remote monitoring through a web-based dashboard.

Together, these modules form a modular software architecture that supports reliable, edge-based power monitoring.

3.4.1 Data Acquisition Engine

The data acquisition layer is implemented by the `NesnasCoreEngine` component within the `nesnas_core.py` module. This subsystem is responsible for continuous communication with the ADS1256 ADC through the SPI interface, synchronization with the DRDY signal, and collection of the raw voltage and current samples required for subsequent processing.

While the ADS1256 supports a theoretical maximum sampling rate of 30,000 SPS, operating-system scheduling overhead, SPI transaction latency, and concurrent software tasks limit the practical continuous acquisition rate on the Raspberry Pi platform to approximately 1,800 SPS. At this rate, the processor must acknowledge the DRDY signal and retrieve a new 24-bit conversion approximately every 555 μs. Excessive delays in servicing the converter may result in missed samples and temporal misalignment between voltage and current waveforms, directly affecting the accuracy of the extracted electrical features.

Although lower than the theoretical converter limit, the achieved acquisition rate remains sufficient to capture the 50 Hz mains waveform together with the harmonic content required by the proposed NILM feature extraction pipeline. The acquired samples are subsequently forwarded to the digital signal processing stage, where they are transformed into electrical and spectral descriptors suitable for appliance identification.

3.4.2 Multithreaded Processing Architecture

The Nesnas OS was implemented in Python and therefore operates under the constraints imposed by the Global Interpreter Lock (GIL). To prevent computationally intensive operations from interfering with time-critical data acquisition, a concurrent architecture was developed that separates hardware polling and signal processing from communication, storage, and visualization services

The core of the architecture is the `NesnasCoreEngine`, implemented within the `nesnas_core.py` module as a dedicated daemon thread. This component maintains continuous communication with the ADS1256 through the SPI interface and executes the complete DSP pipeline for each acquisition window. The processing chain includes DC-offset removal, calibration scaling, frequency estimation, RMS computation, active and reactive power estimation, power-factor calculation, harmonic analysis, THD extraction, and temporal smoothing. The resulting telemetry packet contains both time-domain and frequency-domain descriptors, including electrical quantities, power-quality indicators, and harmonic-spectrum information required for subsequent analysis.

Processed telemetry is transferred through a bounded thread-safe `queue.Queue()` structure. This queue acts as a synchronization boundary between the acquisition engine and the remaining software services, ensuring that delays caused by logging, visualization, networking, or inference tasks cannot propagate back into the sampling loop.

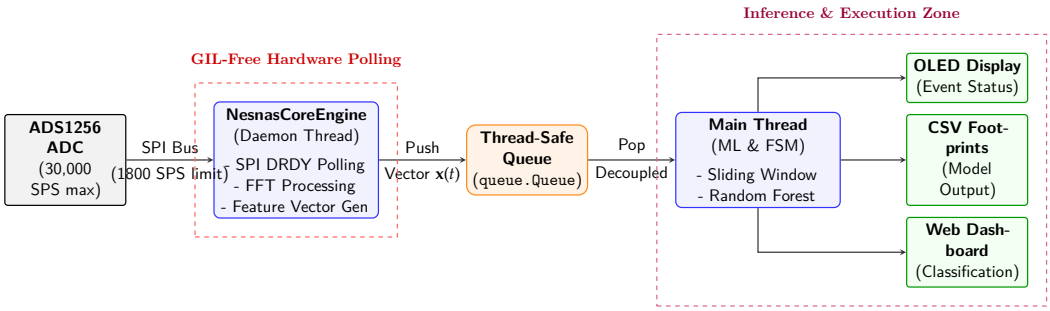


Figure 3.10: Bifurcated software architecture.

3.4.3 Data Routing and System Services

Once telemetry packets are generated by the acquisition engine, they are consumed by a dedicated routing layer implemented within the `nesnas_os.py` module. This component acts as the central coordination service of the Nesnas operating system, distributing measurement data to the local user interface, persistent storage mechanisms, and remote monitoring services. In addition to telemetry management, the routing layer periodically collects system-health indicators, including processor utilization, memory consumption, and processor temperature, enabling continuous monitoring of the embedded platform.

To prevent communication overhead from affecting system responsiveness, telemetry broadcasts are rate-limited before transmission through the WebSocket interface. This approach decouples dashboard refresh operations from the acquisition process, allowing continuous measurement and processing to proceed independently of network latency or browser rendering performance. Consequently, real-time monitoring can be maintained without compromising the timing integrity of the underlying data acquisition pipeline.

3.4.4 Local User Interface

Local interaction is provided through a dedicated hardware interface layer implemented within the `nesnas_ui.py` module. This subsystem manages the 2.42-inch

OLED display and GPIO-connected tactile button array, providing direct access to system telemetry without requiring external devices or network connectivity.

The interface presents multiple information views, including electrical quantities, power-quality indicators, harmonic spectra, and system diagnostics. Navigation is performed through a set of GPIO-connected pushbuttons supporting manual page selection and automatic screen rotation. In addition, a long-press shutdown mechanism was implemented to allow safe system termination directly from the device enclosure.

The user interface executes as an independent background thread and continuously retrieves the latest telemetry data from shared memory maintained by the routing layer. By isolating display rendering and input handling from the acquisition engine, user-interface operations do not interfere with the timing requirements of data acquisition or signal processing, thereby preserving measurement integrity during continuous operation.

3.4.5 Remote Monitoring Dashboard

To provide remote access to system telemetry, a web-based monitoring dashboard was developed using the `nesnas_web.py` backend service and a browser-based frontend implemented in HTML, JavaScript, and Chart.js. The dashboard receives real-time telemetry packets through WebSocket communication, allowing continuous visualization of measurement data without requiring direct interaction with the embedded platform.

The interface provides multiple visualization panels, including a real-time voltage and current oscilloscope, active-power trend monitoring, harmonic spectrum analysis, and a graphical power triangle representing active, reactive, and apparent power. In addition to electrical measurements, the dashboard displays system-health information such as processor temperature, processor utilization, memory consumption, and effective sampling rate.

The dashboard also incorporates a dataset acquisition interface that allows appliance recordings to be labeled and stored directly from the browser. This functionality was used during dataset construction and model development, providing an integrated environment for data collection, system monitoring, and performance evaluation.



Figure 3.11: Macroscopic overview of the developed web dashboard interface for the Air Conditioner node.

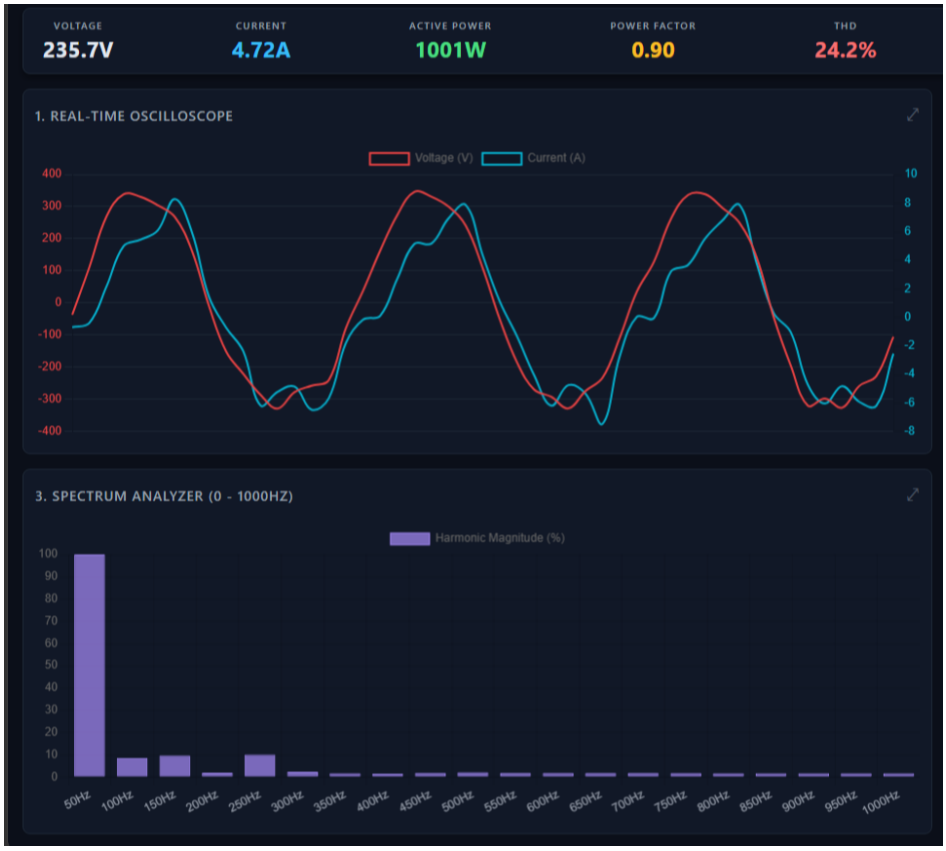


Figure 3.12: Detailed metrology analysis panels for the Air Conditioner node: left panel view showing real-time power metrics and waveform shapes.



Figure 3.13: Detailed metrology analysis panels for the Air Conditioner node: right panel view showing harmonic distribution array and logging.

Figures 3.11 illustrate the web-based monitoring dashboard used for remote observation of system operation. The dashboard aggregates electrical measurements, signal-analysis outputs, and system-health indicators into a single interface, enabling operators to monitor appliance behavior and verify acquisition performance in real time. During dataset collection, the integrated recording interface was used to label and store appliance measurements directly from the browser, simplifying the process of building training datasets for machine-learning applications.

3.5 Feature Extraction Pipeline

The feature extraction pipeline implemented in `nesnas_core.py` transforms raw voltage and current measurements into a feature vector, $\mathbf{x}(t)$, which serves as the input to the appliance-classification model. The objective of this stage is to reduce high-rate waveform data into a compact set of electrical descriptors that capture both steady-state power characteristics and spectral signatures of connected loads.

3.5.1 Signal Preprocessing

Voltage and current samples are continuously acquired from the ESP32-S3 analog-to-digital converter and stored in fixed-length acquisition windows. Because both sensing channels are biased around half of the ADC reference voltage, a DC-offset removal stage is first applied to center the waveforms around zero:

$$v_c[n] = v[n] - \bar{v} \tag{3.1}$$

$$i_c[n] = i[n] - \bar{i} \tag{3.2}$$

where $\bar{v}$ and $\bar{i}$ denote the mean values of the sampled voltage and current arrays, respectively. The resulting centered waveforms are then used for all subsequent calculations.

3.5.2 Time-Domain Feature Computation

The first set of features describes the fundamental electrical behavior of the appliance in the time domain. Root Mean Square (RMS) voltage and current are calculated according to

$$V_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} v_c[n]^2} \tag{3.3}$$

$$I_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} i_c[n]^2} \tag{3.4}$$

where N is the number of samples within the acquisition window.

The instantaneous power waveform is obtained from the point-wise product of the voltage and current signals,

$$p[n] = v_c[n], i_c[n] \tag{3.5}$$

from which the average active power is computed as

$$P = \frac{1}{N}\sum_{n=1}^{N} p[n]. \tag{3.6}$$

The apparent power is then determined as

$$S = V_{RMS} I_{RMS} \tag{3.7}$$

and the reactive power is estimated using the power-triangle relationship

$$Q = \sqrt{S^2 - P^2}. \tag{3.8}$$

The power factor is subsequently calculated as

$$PF = \frac{P}{S}. \tag{3.9}$$

These quantities provide a compact representation of the appliance's steady-state electrical demand and energy-conversion characteristics.

3.5.3 Frequency and Harmonic Analysis

To characterize waveform distortion and non-linear load behavior, frequency-domain analysis is performed using a Fast Fourier Transform (FFT) applied to the acquired current signal. The dominant spectral component is used to estimate the mains frequency, while harmonic magnitudes are extracted from integer multiples of the fundamental frequency.

The Total Harmonic Distortion (THD) is computed as

$$THD = \frac{\sqrt{\sum_{h=2}^{H} I_h^2}}{I_1} \times 100 \tag{3.10}$$

where I_1 represents the fundamental current component and I_h denotes the magnitude of the h-th harmonic. Harmonic amplitudes and THD provide discriminative information for distinguishing between appliances with similar power consumption but different internal electrical topologies.

3.5.4 Feature Vector Construction

The extracted electrical descriptors are combined into a feature vector

$$\mathbf{x}(t) = [V_{RMS}, I_{RMS}, P, Q, S, PF, f, THD, H_2, H_3, \ldots, H_n] \tag{3.11}$$

where f denotes the estimated mains frequency and H_k represents the normalized magnitude of the k-th harmonic component. The resulting feature vector summarizes the instantaneous electrical signature of the connected appliance and is transmitted to the machine-learning inference engine for appliance identification.

Unlike approaches that attempt to compensate for sensor-induced phase offsets through software correction, the NESNAS framework preserves the measured waveform relationships. Any fixed phase displacement introduced by the sensing chain remains deterministic across measurements and therefore becomes part of the appliance's observed electrical signature. During training, the Random Forest classifier learns these consistent characteristics directly from the data, improving robustness under real-world operating conditions.

3.6 Local Human-Machine Interface (HMI) and Edge Navigation

To facilitate autonomous Edge-AI data collection without requiring a continuous Secure Shell (SSH) terminal connection to the workstation, the physical HMI was programmed with a state-machine navigation logic. This allows the operator to control the execution threads, monitor live telemetry, and trigger the ground-truth labeling process entirely from the embedded device.

As illustrated in the Computer-Aided Design (CAD) topography (Figure 3.14), the front panel features a four-button tactile array positioned adjacent to the OLED display. These buttons bypass the standard OS scheduler, utilizing physical GPIO edge-detection to trigger hardware interrupts within the `nesnas_ui.py` script.



Figure 3.14: CAD layout of the front panel HMI.

The specific execution logic allocated to each tactile button is defined as follows:

- **Button 1 (Live Metrology Dashboard):** Commands the OLED to render the real-time telemetry array (Figure 3.15). It actively queries the asynchronous queue to display the True RMS Voltage (V), RMS Current (I), Active Power (P), Power Factor (PF), THD, and the precise SPI polling speed (SPS).
- **Button 2 (ML Studio Initialization):** Transitions the OS from passive monitoring into the ML Studio data collection mode (Figure 3.16). It prepares the local file system to buffer a new continuous Comma Separated Values (CSV) log, maintaining the system in an ”IDLE” state until the operator initiates appliance actuation.
- **Button 3 (Recording Toggle):** Operating as a state-toggle, this initiates the localized recording thread (Figure 3.17). The 1 SPS feature vector $\mathbf{x}(t)$ is continuously written to the Secure Digital (SD) card. A secondary press terminates the recording block, safely closing the CSV file to prevent memory corruption.
- **Button 4 (System Halt):** Executes a safe, programmatic OS-level shutdown (Figure 3.18), closing all SPI ports and halting the AI threads before removing the 230 V power supply to prevent file system corruption.

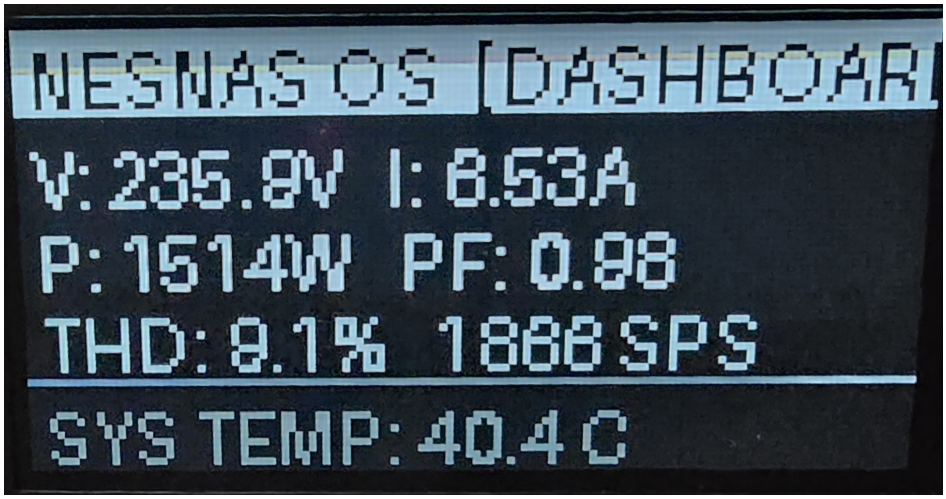


Figure 3.15: Button 1 logic displaying the live metrology dashboard.

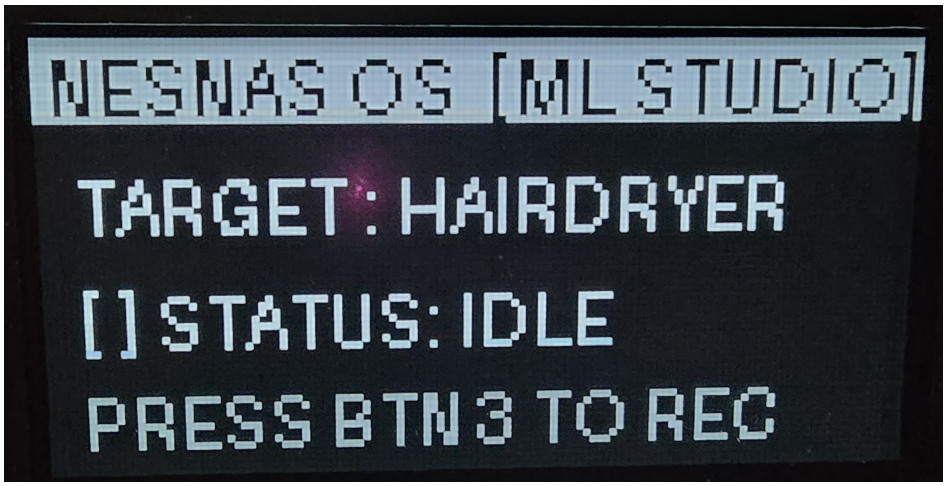


Figure 3.16: Button 2 logic demonstrating the system entering ML Studio mode.

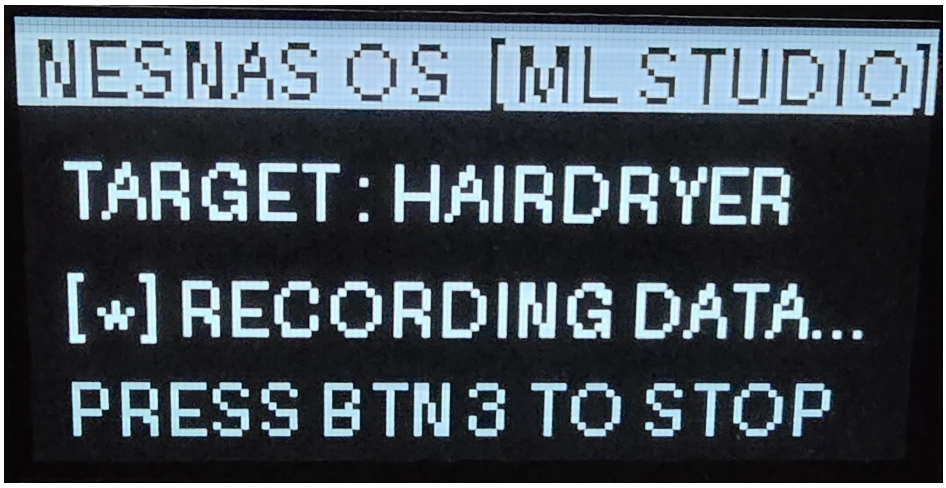


Figure 3.17: Button 3 logic indicating the activation of the physical recording sequence.



Figure 3.18: Button 4 logic initiating a secure, programmatic termination of the Linux kernel to protect the file system.

3.7 Chapter Summary

This chapter outlined the transition from hardware architecture to a functional, localized Edge computing instrument. Mechanical iterations were implemented to ensure safe component isolation and optimal thermal management for the Cortex-A72 processor. Crucially, a bifurcated software architecture was developed to mitigate the constraints of the Python GIL, ensuring that the continuous 1,800 SPS hardware polling loop remains uninterrupted during Random Forest inference. The DSP pipeline efficiently transforms raw ADC telemetry into a structured feature vector $\mathbf{x}(t)$, forming the computational foundation for the classification and validation models presented in the subsequent chapter.

Chapter 4

Experimental Evaluation and Appliance Classification

4.1 Introduction

This chapter presents the experimental evaluation of the proposed Edge-NILM architecture and the development of the appliance classification framework. The objective is to assess the ability of the system to acquire electrical measurements, generate discriminative appliance features, train a machine-learning classifier, and perform real-time appliance identification on the target embedded platform.

The chapter begins by introducing the machine-learning concepts relevant to appliance classification, with particular emphasis on supervised learning and the Random Forest algorithm adopted in this work. The experimental methodology used to construct the appliance dataset is then presented, including the test bench configuration, appliance selection process, data acquisition procedure, and measurement validation experiments. Following dataset preparation and feature engineering, the model training workflow and validation strategy are described. Finally, the chapter presents the classification results obtained during offline training and evaluates the performance of the deployed model within the real-time inference framework operating on the Raspberry Pi platform.

4.2 Machine Learning Background

Machine learning provides computational methods that enable systems to learn patterns directly from data and make predictions without being explicitly programmed for every possible scenario. In classification problems, a model is trained using examples whose correct labels are known and subsequently used to predict the class of previously unseen observations. The ability of machine learning algorithms to identify complex relationships within multidimensional feature spaces has led to their widespread adoption in signal processing, pattern recognition, computer vision, and energy analytics applications. In the context of appliance identification, machine learning models learn the relationship between extracted electrical features and appliance classes, allowing switching events to be automatically classified based on their measured characteristics. This section introduces the supervised learning approach adopted in this work and provides a brief overview of the classification algorithms evaluated during the experimental

study.

4.2.1 Supervised Learning

Supervised learning is a machine learning paradigm in which a model is trained using a set of labeled observations. Each training sample consists of an input feature vector and a corresponding target label representing the correct class. The objective of the learning process is to construct a predictive function capable of mapping previously unseen feature vectors to their associated classes.

A supervised classification dataset can be represented as

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \tag{4.1}$$

where x_i denotes the input feature vector and y_i represents the corresponding class label.

During training, the classifier learns decision boundaries that separate observations belonging to different classes. Once training is completed, the model is evaluated using unseen data to assess its ability to generalize beyond the training dataset. Supervised learning is particularly suitable for appliance identification problems because experimentally acquired electrical events can be labeled according to the appliance that generated them, allowing the classifier to learn discriminative appliance signatures [3, 10].

4.2.2 Random Forest

Random Forest is an ensemble learning algorithm introduced by Breiman. The method constructs a collection of decision trees instead of relying on a single tree. Each tree is trained using a randomly selected subset of training samples and predictor variables, creating diversity within the ensemble.

During classification, each decision tree independently predicts the class of the input feature vector. The final prediction is determined through majority voting among all trees:

$$\hat{y} = \text{mode}\,(T_1(x), T_2(x), \ldots, T_N(x)) \tag{4.2}$$

where $T_i(x)$ is the prediction of the i^{th} decision tree and N is the total number of trees within the forest.

The ensemble voting mechanism reduces the variance typically associated with individual decision trees and improves generalization performance. Random Forest classifiers are also relatively resistant to overfitting, require minimal feature scaling, and can effectively model nonlinear relationships between predictor variables and target classes.

These characteristics make Random Forest particularly attractive for appliance identification applications, where electrical features may exhibit complex interactions and non-linear decision boundaries.

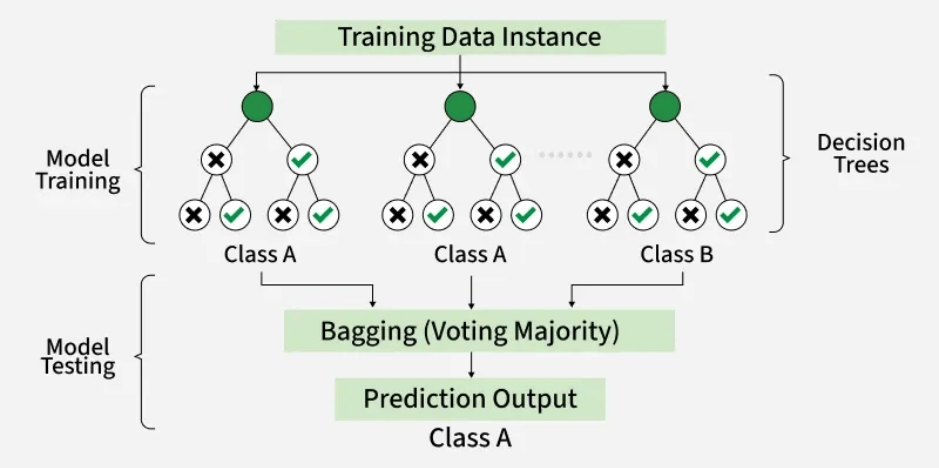


Figure 4.1: Conceptual illustration of the Random Forest classification process. [1].

As illustrated in Figure 4.1, each decision tree generates an independent classification decision. The ensemble prediction corresponds to the class receiving the highest number of votes across all trees.

4.2.3 Comparative Classification Models

To evaluate the suitability of different machine-learning approaches for appliance identification, five supervised classification algorithms were investigated:

- Random Forest
- Extreme Gradient Boosting (XGBoost)
- Light Gradient Boosting Machine (LightGBM)
- Support Vector Machine Classifier (SVM-C)
- K-Nearest Neighbors (KNN)

Random Forest is an ensemble-learning method based on bagging and majority voting among multiple decision trees. XGBoost and LightGBM are gradient-boosting frameworks that construct trees sequentially in order to minimize classification errors. The SVM classifier constructs decision boundaries by maximizing the margin between appliance classes within a transformed feature space. KNN is a non-parametric method that classifies observations based on the majority class among the k nearest neighbors in the feature space.

All classifiers were trained using the same feature set and evaluated using an identical validation procedure. Model selection was performed using macro-averaged F1-score obtained through stratified cross-validation. The classifier achieving the highest validation performance was selected for deployment on the Raspberry Pi platform.

4.3 Experimental Methodology

This section describes the experimental procedures used to construct the appliance dataset employed for machine-learning training and evaluation. The methodology

includes the design of the experimental test bench, electrical isolation measures, appliance selection criteria, and the data acquisition procedure used to generate labeled appliance events.

4.3.1 Test Bench

To provide the Random Forest classifier with labeled training data, an experimental test bench was constructed. The core of the test bench consisted of a modified four-port multisocket designed specifically to segregate sensing hardware from load actuation. The bench was used to generate ground-truth appliance events under controlled operating conditions.

A critical design decision was the exclusion of the Raspberry Pi power supply from the measurement circuit. Because the embedded Edge-AI algorithms generate a variable computational load ranging from approximately 3 W to 15 W, powering the platform from the monitored circuit would introduce parasitic interference into the appliance measurements and contaminate the recorded electrical signatures.

To prevent this effect, the Raspberry Pi was powered from a separate wall outlet while the monitored appliances remained connected to the experimental test bench.

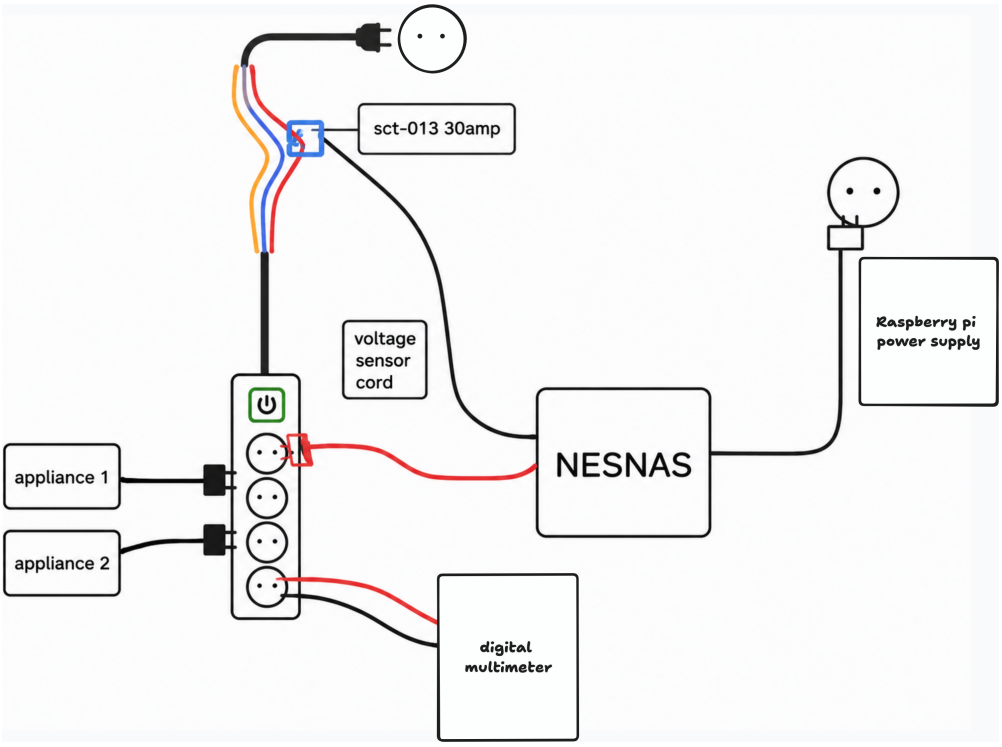


Figure 4.2: Experimental test bench setup.

Figure 4.2 illustrates the electrical configuration used throughout the data acquisition process. By isolating the Raspberry Pi power supply from the measurement circuit, variations in computational load were prevented from influencing the recorded appliance signatures.

4.3.2 Safety and Electrical Isolation

Electrical isolation between the embedded computing platform and the monitored appliance circuit was maintained throughout all experiments. By powering the Raspberry Pi from an independent outlet, variations in computational power consumption were prevented from appearing in the measured electrical features.

This arrangement ensured that the recorded signatures originated solely from the target appliances, thereby preserving the integrity of the ground-truth dataset used for machine-learning training and evaluation.

4.3.3 Appliance Selection

To evaluate the ability of the proposed system to distinguish between different electrical operating conditions, the experimental dataset was constructed using two household appliance categories exhibiting distinct electrical characteristics.

The first category consisted of an electric kettle operating as a predominantly resistive load. This appliance is characterized by a substantial increase in active power consumption while maintaining a power factor close to unity and exhibiting minimal harmonic distortion. Consequently, it provides a representative example of a simple resistive electrical load.

The second category consisted of a household hairdryer. In contrast to the kettle, the hairdryer combines resistive heating elements with a motor-driven fan, introducing both resistive and inductive load components and producing a more complex electrical signature.

Six operating states were considered:

- Cold Power 1
- Cold Power 2
- Warm Power 1
- Warm Power 2
- Hot Power 1
- Hot Power 2

These operating modes generate distinct variations in active power, reactive power, power factor, and harmonic content, thereby providing multiple appliance states for classification and enabling a comprehensive evaluation of the proposed appliance-recognition framework.

The final machine-learning dataset therefore consisted of seven classes: one kettle operating state and six hairdryer operating states.

4.3.4 Acquisition Procedure

Ground-truth data collection was performed using the Nesnas ML Studio interface. For each recording session, the operator manually selected the target appliance state and initiated the recording sequence through the local web dashboard.

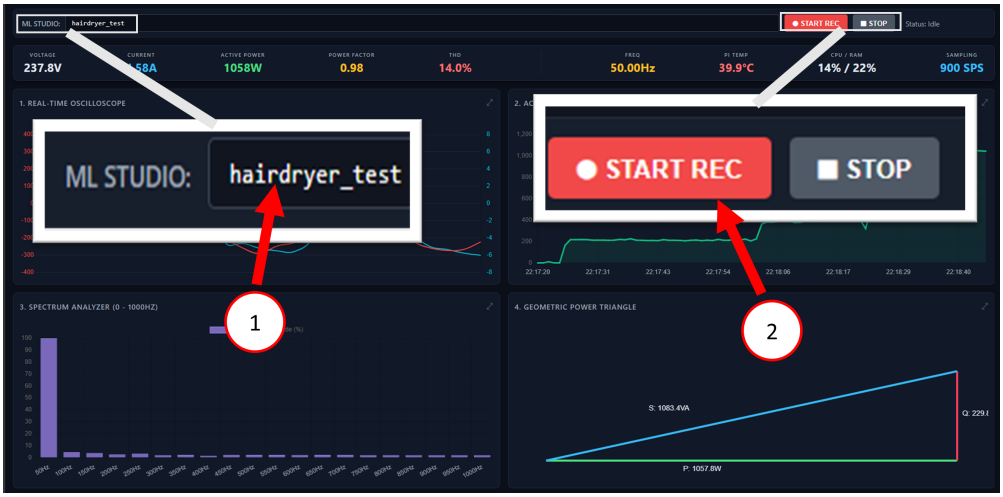


Figure 4.3: Nesnas ML Studio interface during appliance event acquisition.

As shown in Figure 4.3, the acquisition process begins by assigning the correct appliance label within the ML Studio interface. To capture the natural variability of appliance operation, each appliance state was physically switched on and off for 30 consecutive cycles. This procedure was repeated independently for every appliance state included in the dataset.

Although the acquisition engine continuously sampled the ADC at empirical rates between approximately 900 and 1800 samples per second, the logging process stored only synthesized feature vectors at a rate of 1 sample per second. The high-frequency measurements were processed internally by the digital signal processing pipeline to compute electrical metrics and harmonic features. Once the required features were extracted, the raw waveform data was discarded and only the processed feature vectors were retained.

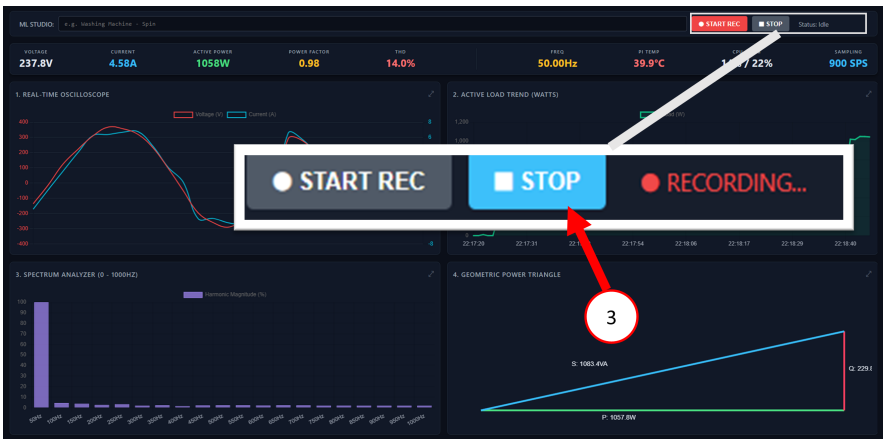


Figure 4.4: Termination of the acquisition session after completion of the recording sequence.

Upon completion of the acquisition session, the recording process was terminated, as illustrated in Figure 4.4. The resulting feature vectors were then exported directly to the local file system as CSV files.

Table 4.1: Excerpt from the exported CSV dataset

Timestamp	V_{RMS}	I_{RMS}	P **(W)**	S **(VA)**	Q **(VAr)**	**PF**	**THD (%)**	H_{50}	H_{100}	**...**	**Appliance_Label**
13:29:31	244.98	0.000	0.0	0.0	0.0	1.000	0.0	100.0	0.00	...	hairDryer_cold_power1
13:29:32	241.95	0.675	144.9	162.6	73.5	0.932	22.6	100.0	15.91	...	hairDryer_cold_power1
13:29:33	242.83	1.084	224.6	263.1	134.6	0.864	38.2	100.0	24.78	...	hairDryer_cold_power1
13:29:35	243.89	1.120	230.7	272.8	141.2	0.855	42.0	100.0	27.39	...	hairDryer_cold_power1

4.3.5 Data Validation and Benchmarking

Prior to constructing the machine-learning dataset, the measurement subsystem was experimentally validated to assess its ability to accurately characterize appliance operating conditions under practical deployment scenarios. The objective of this benchmarking stage was to verify the integrity of the sensing hardware, digital signal processing pipeline, and feature extraction framework before the resulting measurements were used for machine-learning training and real-time appliance classification.

The benchmarking procedure focused on validating the electrical measurements generated by the proposed system and assessing their consistency under both simple and more demanding operating conditions. In addition, the measurements obtained from the platform were compared against an independent reference instrument to verify the accuracy of the implemented metrology framework.

All validation experiments were conducted using the experimental arrangement shown in Figure 4.5. The setup combines the Nesnas monitoring platform, the instrumented multisocket test bench, the ML Studio telemetry interface, and a Fluke 179 True RMS multimeter used for reference verification. This configuration was employed throughout the benchmarking process to evaluate measurement accuracy, signal-processing stability, and feature extraction performance.



Figure 4.5: Experimental setup used during the benchmarking and validation experiments.

Kettle Benchmark Test

The first benchmarking experiment was performed using the kettle appliance introduced in Section 4.3.3. During operation, electrical measurements were continuously acquired through the sensing subsystem and processed by the embedded digital signal processing framework. The resulting telemetry is presented in Figure 4.6.

The recorded measurements exhibited the expected characteristics of a resistive appliance. The voltage and current waveforms remained closely aligned in phase, indicating minimal phase displacement between the two quantities. Consequently, the



Figure 4.6: Macroscopic overview of the web dashboard for the Kettle node.

measured power factor remained close to unity at approximately 0.98. Under steady-state conditions, the active power stabilized at approximately 1514 W, while the apparent power remained closely aligned with the active power, reflecting the predominantly resistive nature of the load.

Frequency-domain analysis further confirmed this behaviour. The harmonic spectrum was dominated by the fundamental 50 Hz component, while higher-order harmonics remained comparatively small. The resulting Total Harmonic Distortion (THD) was measured at approximately 9.1%, indicating that the observed distortion originated primarily from background grid conditions rather than from the appliance itself.

In addition to the remote telemetry interface, a localized embedded dashboard was used to verify the stability of the real-time processing subsystem. The local display shown in Figure 4.8 reported the same electrical quantities presented through the web interface while maintaining a sampling rate of approximately 1866 samples per second.



Figure 4.8: Localized Nesnas OS OLED dashboard during the kettle test.

The agreement between the local and remote telemetry outputs confirms the correct operation of the acquisition pipeline and demonstrates that the embedded processing subsystem was capable of sustaining high-frequency sampling while simultaneously performing real-time electrical analysis.



(a) Left panel view: real-time power metrics and waveform shapes.



(b) Right panel view: harmonic distribution array and Power triangle.

Figure 4.7: Detailed metrology analysis panels for the Kettle node.

Hairdryer Benchmark Test

The second validation experiment evaluated the system response during hairdryer operation using the same experimental arrangement presented in Figure 4.5. The resulting telemetry generated by the sensing and processing framework is shown in Figure 4.9.



Figure 4.9: Macroscopic overview web dashboard interface for the Hairdryer .

The measured current waveform exhibited visible asymmetry and distortion relative to the supply voltage waveform. This behaviour resulted from the interaction between the heating elements and motor assembly and generated a substantially more complex electrical footprint than that observed during the kettle benchmark.

The increased waveform distortion reduced the measured power factor to approximately 0.83. Correspondingly, the power triangle exhibited a larger separation between active and apparent power, with measured values of approximately 209 W active power, 254.6 VA apparent power, and 142.5 VAr reactive power.

Frequency-domain analysis revealed a pronounced second-harmonic component at 100 Hz. This harmonic contribution increased the measured THD to approximately 41.0%, producing a feature space that was substantially different from that observed for the resistive benchmark load. These characteristics demonstrate the ability of the sensing framework to capture nonlinear appliance behaviour and preserve the electrical information required for appliance discrimination.

The stability of the embedded processing subsystem was also evaluated under these more demanding operating conditions. Figure 4.11 presents the localized OLED dashboard during hairdryer operation.



(a) Left panel view: real-time power metrics and waveform shapes.



(b) Right panel view: harmonic distribution array and logging.

Figure 4.10: Detailed metrology analysis panels for the Hairdryer.

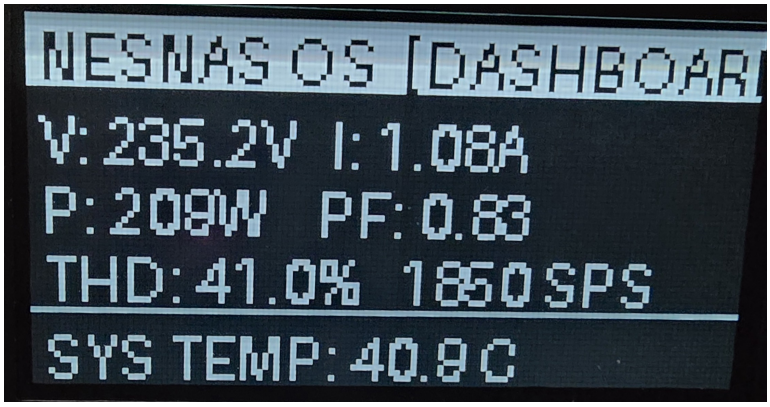


Figure 4.11: Localized Nesnas OS OLED dashboard during the hairdryer test.

Despite the increased computational demand associated with harmonic analysis and real-time feature extraction, the system maintained a sampling rate of approximately 1850 samples per second while operating within acceptable thermal limits. This result demonstrates the stability of the acquisition and processing architecture under nonlinear load conditions.

Calibration Verification

Following the appliance benchmarking experiments, the accuracy of the voltage measurement subsystem was evaluated through comparison with a Fluke 179 True RMS digital multimeter. The reference instrument was used to verify the correctness of the voltage measurements produced by the proposed system under both benchmark conditions.

During the kettle experiment, the Fluke instrument measured a grid voltage of 234.2 V AC, as shown in Figure 4.12.

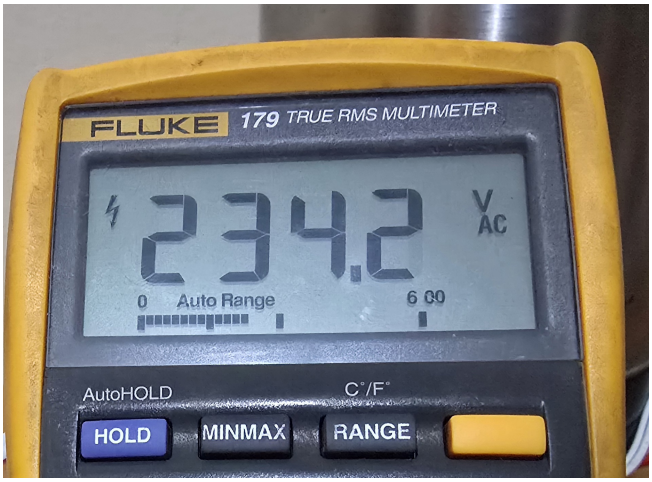


Figure 4.12: Fluke 179 True RMS Multimeter acting as the empirical witness.

At the same operating point, the proposed system reported a voltage of 235.8 V. The resulting discrepancy corresponds to an error below 0.7%, indicating close agreement between the measurement subsystem and the reference instrument.

A second verification was performed during the hairdryer benchmark. Under the distorted load conditions, the Fluke instrument measured a grid voltage of 235.6 V AC, as shown in Figure 4.13.

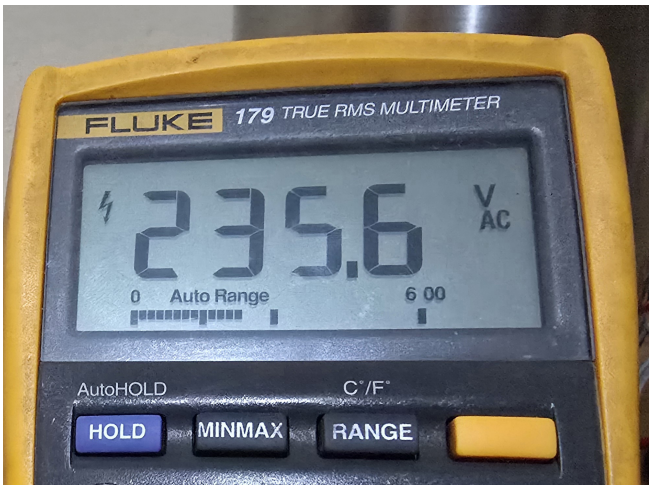


Figure 4.13: Fluke 179 True RMS Multimeter acting as the empirical witness.

Simultaneously, the proposed system reported a voltage of 235.2 V. The resulting discrepancy was 0.4 V, corresponding to an error of approximately 0.16%.

Table 4.2: Comparison between the proposed measurement subsystem and the Fluke 179 True RMS reference instrument during benchmarking experiments.

Benchmark Condition	Fluke 179 (V)	Nesnas Platform (V)	Error
Kettle Benchmark	234.2	235.8	$< 0.7\%$
Hairdryer Benchmark	235.6	235.2	0.16%

Table 4.2 summarizes the voltage measurements obtained from the proposed sensing platform and the Fluke 179 True RMS multimeter during both benchmarking experiments. In both cases, the measured discrepancies remained below 1%, demonstrating close agreement between the two instruments. The low error margins confirm the effectiveness of the implemented voltage measurement and calibration framework and indicate that the sensing subsystem maintains measurement accuracy under both predominantly resistive and nonlinear operating conditions.

4.4 Experimental Dataset

4.4.1 Dataset Preparation

The experimental dataset was constructed using the appliance acquisition framework described in the previous sections. During each recording session, the target appliance state was selected through the Nesnas ML Studio interface and repeatedly actuated under controlled laboratory conditions.

The final dataset consisted of seven appliance classes corresponding to the operating conditions evaluated during this work:

- Kettle
- Hairdryer Cold Power 1
- Hairdryer Cold Power 2
- Hairdryer Warm Power 1
- Hairdryer Warm Power 2
- Hairdryer Hot Power 1
- Hairdryer Hot Power 2

For each appliance state, approximately thirty switching cycles were recorded. During acquisition, the embedded DSP engine continuously processed the incoming voltage and current measurements and generated one processed feature vector per second. The resulting observations were exported as CSV files and subsequently aggregated to form the machine-learning dataset used throughout the experimental study.

Each appliance state was recorded for approximately 4.5 minutes. Since the feature extraction pipeline generated one processed feature vector per second, each acquisition session produced approximately 250–300 labeled observations.

4.4.2 Dataset Statistics

The resulting class distribution was intentionally maintained as uniformly as possible in order to reduce class imbalance during classifier development. Table 4.3 summarizes the distribution of the collected appliance recordings. Figure 4.14 illustrates the relative contribution of each appliance class within the acquisition dataset.

The relatively balanced distribution of observations reduces the likelihood of majority-class bias and provides a suitable foundation for supervised machine-learning training and evaluation.

Table 4.3: Dataset size and class distribution.

Target Class	**Appliance State**	**Samples (N)**	**Distribution**
Class 1	Kettle	306	16.6%
Class 2	Hairdryer Cold Power 1	266	14.5%
Class 3	Hairdryer Cold Power 2	240	13.1%
Class 4	Hairdryer Warm Power 1	262	14.2%
Class 5	Hairdryer Warm Power 2	250	13.6%
Class 6	Hairdryer Hot Power 1	262	14.2%
Class 7	Hairdryer Hot Power 2	253	13.8%
Total	**7 Classes**	**1,839**	**100%**

The distribution of samples among the seven classes is illustrated in Figure 4.14.

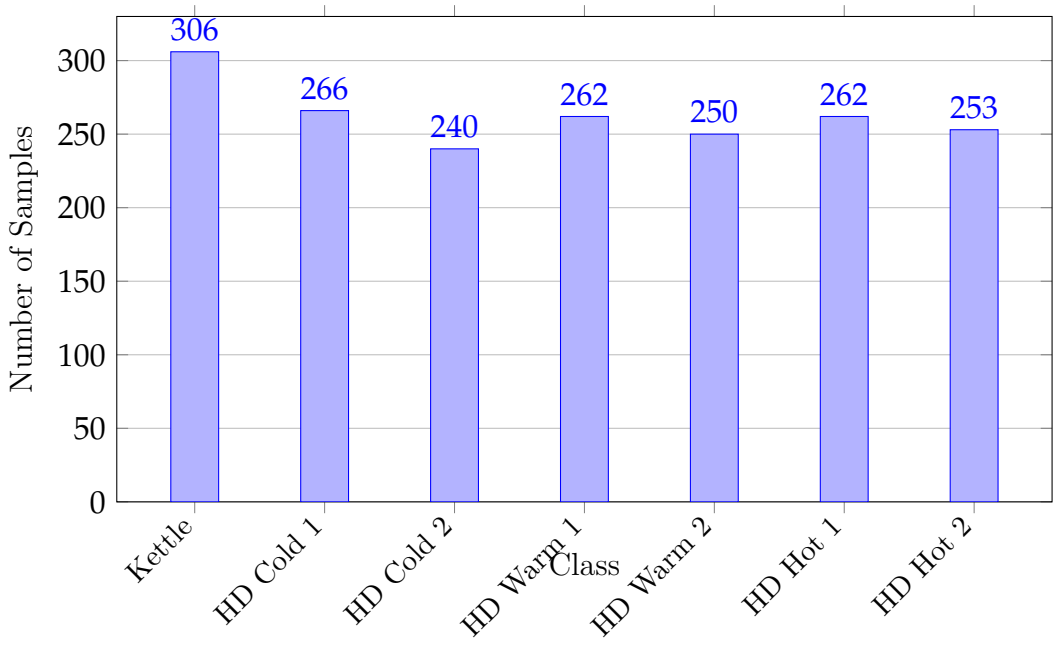


Figure 4.14: Distribution of samples across the seven appliance classes used for machine-learning training and evaluation.

4.5 Feature Engineering

4.5.1 Electrical Feature Extraction

The raw voltage and current measurements acquired during the data collection phase were not used directly for machine-learning training. Instead, the high-frequency telemetry stream generated by the acquisition subsystem was processed by the embedded digital signal processing pipeline to generate a compact feature representation suitable for classification.

Although the acquisition engine continuously sampled the electrical signals at empirical rates ranging from approximately 900 to 1800 samples per second, only processed feature vectors were retained. The high-frequency waveform data served as an intermediate representation used to compute electrical metrics and harmonic information before being discarded.

The resulting feature vector consisted of both time-domain and frequency-domain electrical features. Time-domain quantities included RMS voltage, RMS current, active power, apparent power, reactive power, and power factor. Frequency-domain information was obtained through Fast Fourier Transform (FFT) analysis and included harmonic amplitudes together with Total Harmonic Distortion (THD).

By combining electrical power measurements with harmonic information, the resulting feature space captured both the steady-state operating condition and the spectral characteristics of the monitored appliance. Each processed feature vector was recorded at a rate of one sample per second and stored as a single observation within the dataset.

4.5.2 Temporal Window Generation

Although the acquisition framework generated one processed feature vector per second, appliance operating conditions exhibit temporal behaviour that may not be fully represented by a single observation. Consequently, a temporal sliding-window approach was adopted.

A three-second window was constructed by grouping three consecutive feature vectors. For every electrical feature contained within the window, five statistical descriptors were computed:

- Mean
- Standard deviation
- Minimum value
- Maximum value
- Temporal difference between the first and last observation

This process transformed each three-second observation window into a higher-dimensional feature representation containing both instantaneous electrical measurements and short-term temporal dynamics.

The resulting temporal representation improves robustness against measurement noise and transient fluctuations while preserving appliance-specific operating characteristics.

4.5.3 Data Augmentation

Residential electrical systems are subject to voltage fluctuations caused by load variations and grid conditions. To improve classifier robustness under such conditions, a data augmentation strategy was implemented.

Synthetic voltage-sag conditions were generated using scaling factors of 1.00, 0.95, 0.90, and 0.85 relative to nominal voltage [26]. Voltage and current measurements were scaled linearly, while active, apparent, and reactive power measurements were scaled according to quadratic power relationships derived from Ohm's law.

Power factor, total harmonic distortion, and harmonic amplitudes were preserved to maintain the spectral characteristics of the original appliance signatures.

By exposing the classifier to multiple operating voltages during training, the augmentation process encourages the model to learn appliance-specific electrical characteristics rather than relying solely on absolute power levels.

4.5.4 Training and Testing Dataset Generation

To evaluate classifier performance under realistic operating conditions, the dataset was partitioned using a temporally separated train-test strategy.

For each appliance recording, the first 80% of observations were assigned to the training dataset, while the remaining 20% were reserved for testing. A purge interval equal to the temporal window length was inserted between both subsets to prevent overlapping observations from appearing in both partitions.

Only the training subset was subjected to voltage-sag augmentation. The testing subset remained unchanged and therefore represented genuine unseen appliance measurements.

Following temporal-window generation and augmentation, the final machine-learning dataset consisted of 5,188 training windows and 294 testing windows.

4.6 Model Training

While the Raspberry Pi 4 is capable of executing machine-learning inference, training a Random Forest classifier requires significantly greater computational resources. Consequently, a bifurcated deployment strategy was adopted. The datasets generated by the Edge acquisition platform were securely transferred through SSH to a high-performance workstation where model training was performed.

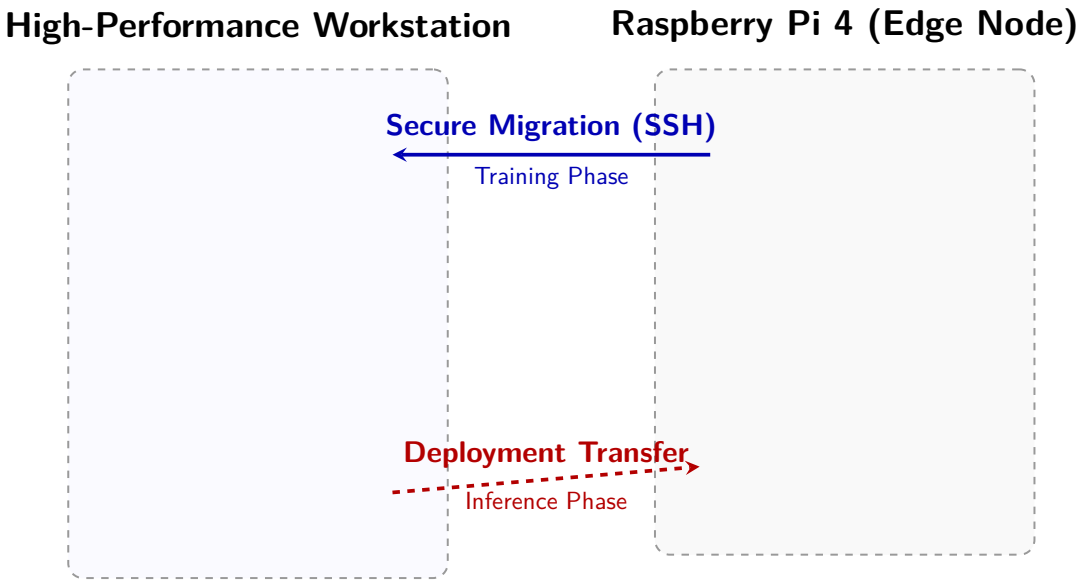


Figure 4.15: Cyclic machine learning workflow.

As illustrated in Figure 4.15, the proposed workflow separates data acquisition from model training. This approach allows the resource-constrained Raspberry Pi to focus on data collection and inference tasks while computationally intensive training operations are executed on a dedicated workstation.

The training procedure was implemented using the `scikit-learn` library [19]. The Random Forest classifier was configured using 300 decision trees and trained using the processed appliance dataset.

The classifier was trained using the training subset generated through the temporally separated partitioning strategy described in Section 4.5.4.

Following convergence, the trained classifier was serialized and exported as the deployment model `nesnas_brain_windowed.pkl`. The resulting model was subsequently transferred back to the Raspberry Pi platform for live deployment.

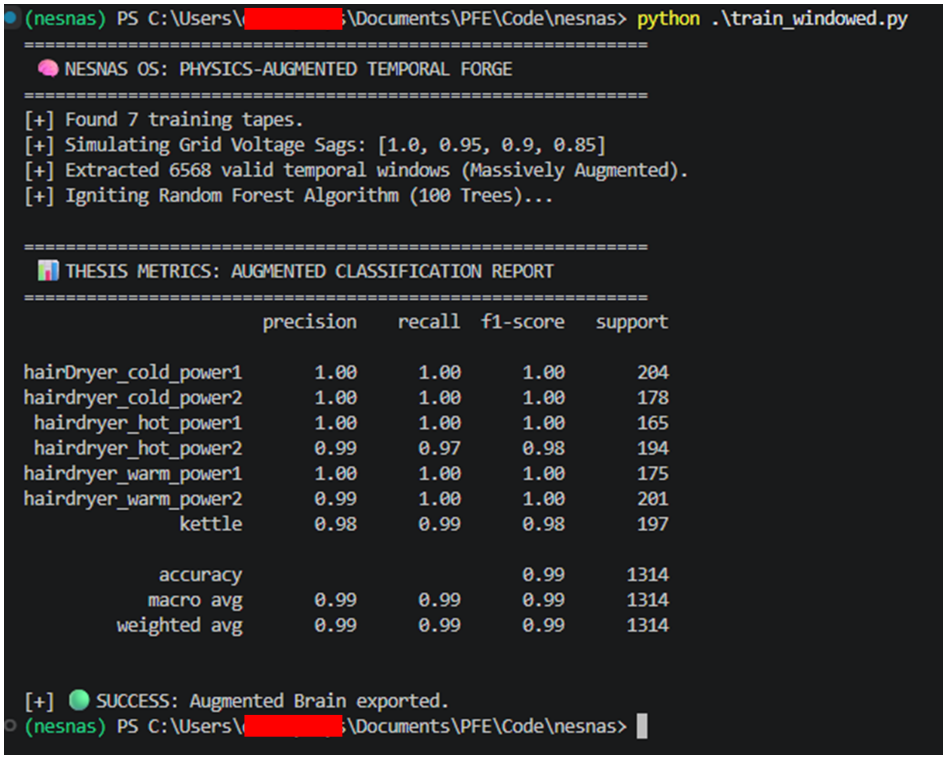


Figure 4.16: Offline training output obtained on the workstation after training the Random Forest classifier on the augmented dataset.

The outcome of the offline training process is shown in Figure 4.16. The figure presents the classification metrics generated by the Random Forest model after training on the augmented dataset, demonstrating the model's ability to distinguish between the appliance classes included in the study.

4.6.1 Validation Strategy

To obtain a reliable estimate of classifier performance, model evaluation was performed using a two-stage validation procedure.

The dataset was first partitioned into temporally separated training and testing subsets, as described in Section 4.5.4. The testing subset was isolated from all training operations and reserved exclusively for final performance evaluation.

Model selection was performed using stratified five-fold cross-validation applied exclusively to the training dataset. During each fold, the classifier was trained on four partitions and validated on the remaining partition. Classification performance was quantified using overall accuracy and macro-averaged F1-score.

Following cross-validation, the classifier achieving the highest macro-averaged F1-score was selected as the deployment candidate. The selected model was then retrained using the complete training dataset and subsequently evaluated on the held-out test set.

This validation methodology provides an unbiased estimate of classification performance while reducing the risk of information leakage between training and testing observations.

4.7 Experimental Results

This section presents the results obtained during offline model training and subsequent deployment of the trained classifier on the target Edge platform.

4.7.1 Classifier Performance Comparison

Table 4.4 summarizes the cross-validation performance obtained by the five evaluated classifiers.

Table 4.4: Cross-validation performance of evaluated classification models.

Model	Accuracy	Macro F1
Random Forest	99.87%	99.86%
XGBoost	99.79%	99.79%
LightGBM	99.61%	99.62%
SVM-C	99.63%	99.64%
KNN	97.13%	97.12%

The selected Random Forest classifier was subsequently evaluated using the temporally separated hold-out dataset. The resulting classification accuracy reached 97.62%, while the macro-averaged F1-score achieved 97.60%. Only five misclassifications were observed among the 294 test samples.

These results demonstrate strong generalization performance despite the use of temporally separated training and testing data. The relatively small gap between cross-validation and test performance indicates that the model successfully learned appliance-specific electrical signatures rather than memorizing individual observations.

Random Forest achieved the highest macro-averaged F1-score among all evaluated classifiers and was therefore selected as the deployment model. Although XGBoost, LightGBM, and SVM achieved comparable performance, Random Forest provided the best overall balance between classification accuracy, model interpretability, and deployment simplicity.

4.7.2 Confusion Matrix Analysis

Figure 4.17 presents the confusion matrix obtained from the Random Forest classifier using the temporally separated test dataset.

The matrix exhibits a dominant diagonal structure, indicating that the vast majority of appliance observations were classified correctly. Out of 294 test samples, only five misclassifications were observed, corresponding to an overall classification accuracy of 97.62%.

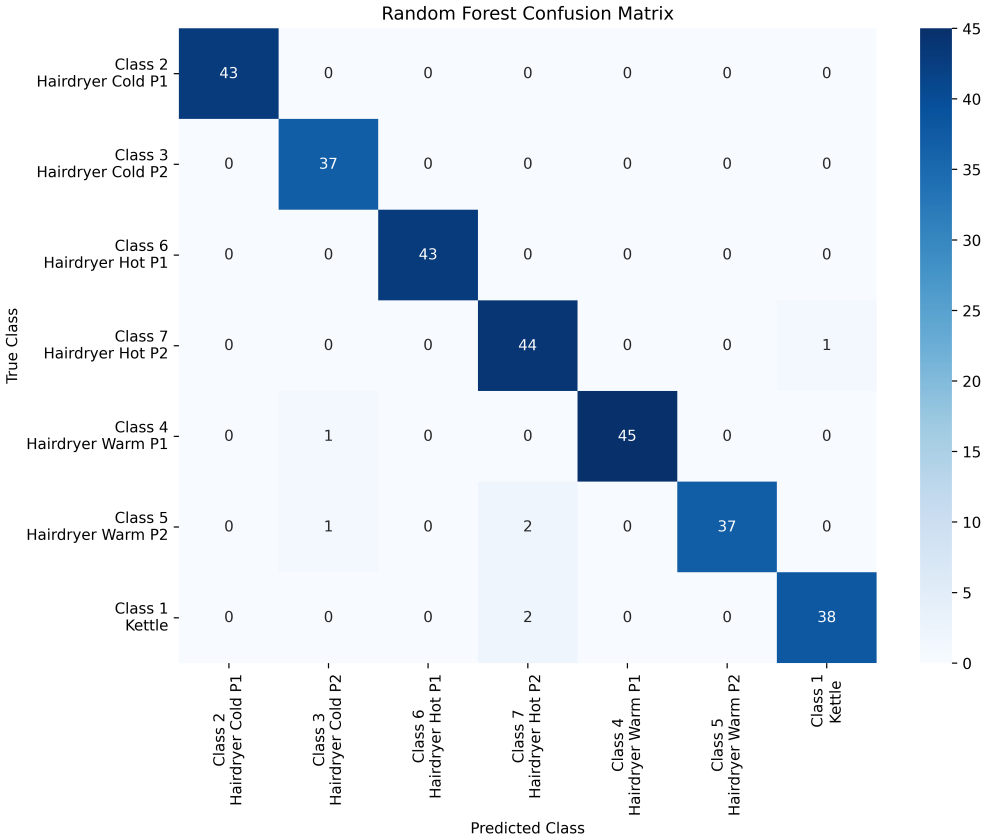


Figure 4.17: Confusion matrix of the Random Forest classifier evaluated on the held-out test set.

Perfect classification was achieved for the Hairdryer Cold Power 1, Hairdryer Cold Power 2, and Hairdryer Hot Power 1 operating states. The remaining classification errors occurred primarily between electrically similar appliance states. In particular, a small number of Hairdryer Warm Power 2 observations were confused with neighboring operating modes, while two kettle observations were incorrectly classified as Hairdryer Hot Power 2.

The observed misclassifications are consistent with the physical characteristics of the appliances. Operating states exhibiting similar active power consumption, RMS current levels, and harmonic content are naturally more difficult to distinguish. Nevertheless, the low number of off-diagonal entries demonstrates that the proposed feature-engineering framework successfully captures the electrical characteristics required for reliable appliance identification.

Overall, the confusion matrix confirms the strong discriminatory capability of the extracted feature set and supports the high classification accuracy achieved by the Random Forest model.

4.8 Real-Time Inference Framework

Following completion of the offline training phase, the serialized model was transferred back to the Raspberry Pi platform. This transformed the device from a passive acquisition instrument into an active Edge-AI appliance classification system.

Live inference was executed through the `live_windowed.py` deployment script. During operation, feature vectors generated by the DSP pipeline were continuously supplied to the trained Random Forest classifier.

A challenge encountered during continuous appliance monitoring is the occurrence of transient switching events. During appliance activation or deactivation, rapidly changing electrical conditions may produce unstable classifications and short-duration false state transitions.

To mitigate this effect, the Nesnas operating system implemented a three-second temporal sliding window governed by a finite-state machine. Instead of evaluating each one-second feature vector independently, the classifier operated on a rolling three-second buffer. A state transition was accepted only when the classification output remained stable throughout the entire temporal window.

Although the Random Forest model performed inference with negligible computational latency, the finite-state machine intentionally introduced a three-second confirmation interval to ensure that the physical appliance state had stabilized before a classification update was issued.

This temporal validation mechanism suppressed transient classification fluctuations and improved the stability of the real-time appliance recognition process. The resulting architecture enabled reliable appliance classification while maintaining compatibility with the computational constraints of the Raspberry Pi platform.

4.8.1 Live Deployment Validation

Following offline training, the serialized Random Forest model was transferred to the Raspberry Pi platform and deployed within the real-time inference framework. The objective of this experiment was to verify that the trained classifier could operate reliably under live conditions and correctly identify appliance state transitions using feature vectors generated by the embedded digital signal processing pipeline.

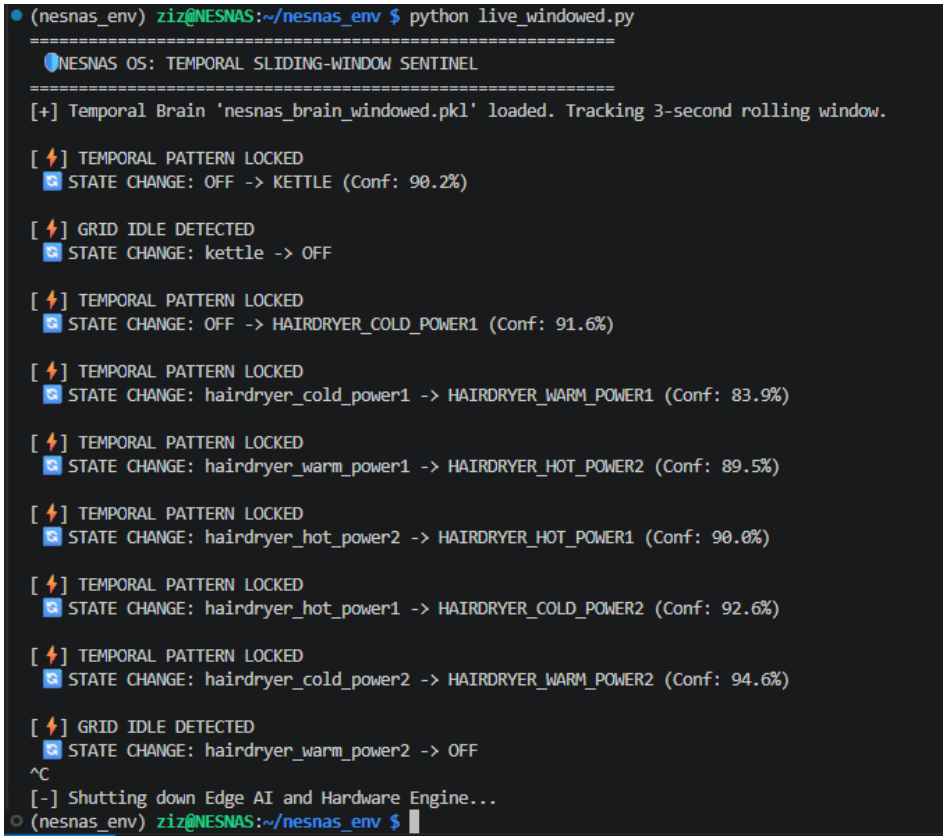


Figure 4.18: Live appliance classification performed by the deployed Edge-AI system using the temporal sliding-window inference framework.

Figure 4.18 presents the console output generated during a live validation session. After loading the deployment model `nesnas_brain_windowed.pkl`, the system continuously processed incoming feature vectors and evaluated appliance states using the temporal sliding-window classifier.

During the experiment, the deployed model successfully detected transitions between the idle condition, kettle operation, and multiple hairdryer operating modes. The system correctly identified state changes including Hairdryer Cold Power 1, Warm Power 1, Hot Power 1, Hot Power 2, Cold Power 2, and Warm Power 2 while simultaneously detecting transitions back to the idle state. These results demonstrate the ability of the deployed classifier to track appliance operating conditions in real time rather than merely performing isolated classifications.

The confidence values reported during live inference generally remained between approximately 84% and 95%, with the majority of classifications exceeding 90%. Furthermore, no unstable oscillations or rapid false transitions were observed during appliance state changes. This behaviour indicates that the temporal validation mechanism successfully suppressed transient classification fluctuations while maintaining responsiveness to genuine appliance events.

Overall, the live deployment experiment confirms that the trained Random Forest model remained functional after migration to the embedded platform and was capable of performing stable real-time appliance identification using the feature vectors generated by the proposed Edge-NILM architecture.

4.9 Chapter Summary

This chapter presented the complete experimental evaluation of the proposed Edge-NILM architecture and the development of the appliance-classification framework. The study began with the construction of a labeled appliance dataset acquired using the experimental test bench and the embedded data-acquisition subsystem. Benchmarking experiments conducted using representative household appliances demonstrated that the sensing platform was capable of accurately characterizing both predominantly resistive and nonlinear electrical loads. Comparison against a Fluke 179 True RMS multimeter confirmed the accuracy of the implemented metrology framework, with voltage measurement errors remaining below 1

The collected measurements were subsequently processed through the embedded digital signal processing pipeline to generate engineered feature vectors containing both time-domain and frequency-domain information. Additional preprocessing stages were introduced, including temporal window generation, statistical feature extraction, physics-based voltage-sag augmentation, and temporally separated train-test partitioning. These steps transformed the raw appliance recordings into a machine-learning dataset suitable for robust appliance classification.

To evaluate the suitability of different machine-learning approaches, five supervised classification algorithms were investigated: Random Forest, XGBoost, LightGBM, Support Vector Machines, and K-Nearest Neighbors. Model selection was performed using stratified five-fold cross-validation applied exclusively to the training dataset. Among the evaluated models, Random Forest achieved the highest macro-averaged F1-score and was therefore selected as the deployment classifier.

Evaluation on the temporally separated hold-out test dataset yielded a classification accuracy of 97.62% and a macro-averaged F1-score of 97.60%. Analysis of the confusion matrix revealed that only five misclassifications occurred among 294 test samples, with the majority of errors occurring between electrically similar hairdryer operating modes. These results demonstrate strong generalization performance and confirm that the extracted feature set successfully captures the electrical characteristics required for reliable appliance identification.

Following offline training, the serialized classifier was deployed on the Raspberry Pi platform and integrated within a real-time inference framework based on a three-second temporal sliding window and finite-state machine. Live deployment experiments confirmed that the system was capable of performing stable appliance recognition in real time while suppressing transient classification fluctuations and maintaining compatibility with the computational constraints of the embedded hardware.

Overall, the results presented in this chapter validate the proposed Edge-NILM architecture and demonstrate the feasibility of combining embedded signal processing, machine-learning classification, and edge computing techniques to perform accurate appliance identification directly on resource-constrained platforms. The next chapter presents the overall conclusions of this work and discusses potential directions for future research and system enhancement.

General Conclusion and Future Perspectives

The realization of the Smart Grid and the optimization of Demand-Side Management (DSM) are highly dependent upon the accuracy, latency, and privacy of electrical load disaggregation. Historically, the pursuit of high-fidelity NILM has driven the academic community toward computationally intensive, cloud-reliant Deep Learning (DL) models. While theoretically accurate, these continuous architectures suffer from systemic latency, bandwidth saturation, and the inherent "sliding smear" error during transient state changes.

This thesis addresses these limitations by proposing, engineering, and validating a localized, **Event-Based Hardware-in-the-Loop** methodology. We designed and deployed "Nesnas"—an autonomous Edge-AI metrology instrument—demonstrating that deterministic machine learning classification can be executed locally at the point of sensing without reliance on external cloud infrastructure.

Synthesis of Contributions

The success of the Nesnas architecture was achieved through a comprehensive engineering approach, bridging the gap between isolated analog metrology and high-level artificial intelligence:

- **High-Fidelity Metrology Realization:** We engineered a custom AFE paired with an ADS1256 24-bit ADC. By mathematically enforcing a 1.65 V DC unipolar bias and integrating hardware-level RC anti-aliasing filters, the system effectively captured phase-coherent, microvolt-level grid transients that standard microcontrollers fail to resolve.
- **Bifurcated Edge Concurrency:** To bypass the severe latency induced by the Python GIL and OS overhead, a multi-threaded software architecture was deployed. By isolating the hardware polling within a high-priority daemon thread, the instrument maintained a stabilized, low-latency acquisition rate exceeding 1,850 SPS even under heavy algorithmic load.
- **Event-Based Random Forest Integration:** We mathematically formulated a deterministic feature space consisting of Active Power (P), Reactive Power (Q), THD, and a 5-bin FFT harmonic array. By buffering this vector into a 3-second Temporal Sliding-Window governed by an FSM, the Random Forest classifier effectively mitigated transient flickering, triggering inference exclusively upon stabilized physical state changes.

Empirical Validation

The algorithmic intelligence was physically validated against a heavily controlled 4-port multisocket test bench. The empirical data demonstrated the system's capability to reliably distinguish between a large baseline resistive step-function (a 1514 W electric kettle) and a highly complex, multi-state non-linear load (a hairdryer utilizing half-wave rectification).

Crucially, the metrology performance was verified against a Fluke 179 True RMS Multimeter empirical witness, confirming a measurement discrepancy of less than 0.16% even under significant harmonic noise. The Nesnas Edge-AI pipeline successfully tracked the hairdryer through its dynamic mechanical states with confidence intervals exceeding 90%. Furthermore, by augmenting the offline training matrix with simulated grid voltage sags of up to 15%, the localized model proved highly resilient to real-world residential voltage fluctuations, achieving F1-scores of 0.98 to 1.00 across all targeted appliance states.

Future Perspectives

While the Nesnas prototype successfully validates the Hardware-in-the-Loop Edge computing paradigm, it establishes a foundational architecture suitable for extensive future expansion. The subsequent avenues for research include:

- **Deployment of Autonomous Load Disaggregation:** While the initial scope of this research envisioned an end-to-end NILM pipeline including live appliance disaggregation, the severe engineering complexity required to achieve high-fidelity, high-speed metrology necessitated a strategic scope reduction. Having now definitively established and validated the deterministic hardware foundation, the immediate subsequent milestone is the training and deployment of advanced machine learning models to execute fully autonomous load disaggregation atop the extracted feature vectors.

- **Algorithmic Scaling and Appliance Libraries:** Following the deployment of the disaggregation models, future work must focus on expanding the training matrix to encompass a vast library of residential appliances, specifically evaluating overlapping loads (e.g., a refrigerator compressor activating while the microwave is running) to test the mathematical limits of the harmonic feature space.

- **Hardware Miniaturization and Custom Silicon:** Transitioning from the modular Raspberry Pi architecture to a custom, monolithic Printed Circuit Board (PCB). Integrating the processor, the ADC, and the AFE onto a single substrate will substantially reduce Electromagnetic Interference (EMI) and optimize the physical footprint for placement directly inside residential breaker panels.

- **Portable Diagnostics and Wireless Telemetry:** Evolving the stationary Edge architecture into a ruggedized, handheld NILM power analyzer. By decoupling the sensing layer via an isolated wireless current clamp, the instrument would allow field engineers to capture high-fidelity harmonic footprints and perform real-time diagnostic disaggregation without being physically tethered to the mains distribution board.

- **Edge-TPU Acceleration:** Integrating a localized Tensor Processing Unit (TPU) (such as the Google Coral) via PCIe would allow the Nesnas system to execute complex LSTM networks directly on the Edge, merging the sequence-awareness of Deep Learning with the low-latency privacy of the current architecture.
- **Polyphase Industrial Expansion:** Adapting the current single-phase metrology pipeline to support three-phase industrial NILM. This would require synchronizing three simultaneous ADC pipelines to footprint heavy industrial induction motors, expanding the localized Edge computing paradigm into the manufacturing sector.

Ultimately, the Nesnas instrument indicates that the future of Smart Grid intelligence does not reside exclusively in centralized server infrastructures. By fusing high-speed analog hardware directly with optimized, event-driven algorithms, we can contribute to a decentralized energy grid that is intelligent, highly responsive, and structurally secure.

Appendix A

Mechanical CAD Evolution and Enclosure Iterations

The physical realization of the NESNAS platform required iterative mechanical engineering to ensure the safety, stability, and thermal efficiency of the Edge-AI instrument. CAD software was heavily utilized to model the overall geometry and optimize the internal component layout prior to fabrication. Figure A.1 illustrates the progression of these structural designs.

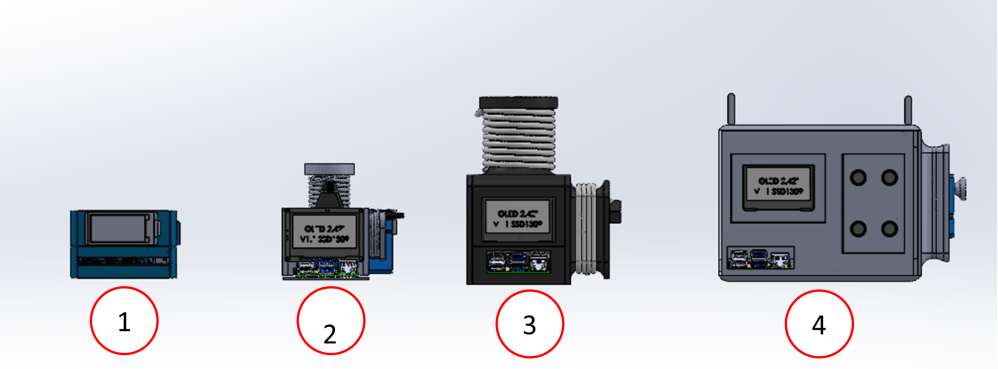


Figure A.1: Mechanical enclosure designs developed for the NESNAS platform. Figures (1) and (2) present preliminary CAD design concepts, while Figures (3) and (4) show the two fabricated enclosure prototypes used during system development.

Prototype 1: Side-Mounted Architecture

Prototype 1 was the first fabricated enclosure, developed primarily to evaluate the practical limitations of 3D printing and assess Design for Manufacturing (DFM) considerations. The design included two removable side panels which served as mounting surfaces for the electronic components.

However, mechanical testing revealed a critical architectural flaw: mounting active electronic components directly on the removable panels introduced severe mechanical strain on the analog conditioning circuits and high-voltage wiring whenever the panels were opened for debugging (see Figures A.2 and A.3). This strain risked compromising circuit stability and measurement reliability.

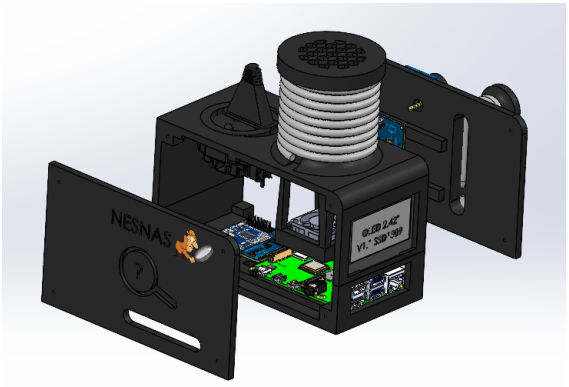


Figure A.2: Left-side view of Prototype 1, illustrating the side-mounted component topology.

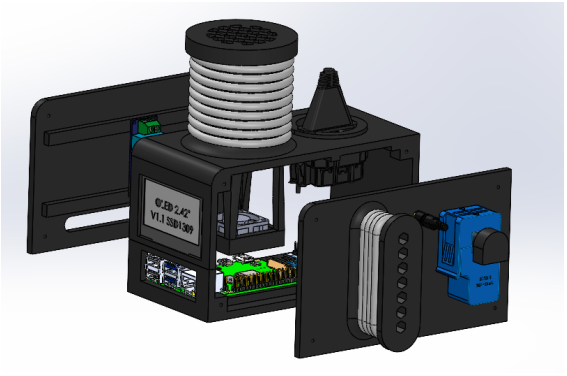


Figure A.3: Right-side view of Prototype 1, demonstrating hazardous strain on internal wiring upon opening.

Prototype 2: Modular Bottom-Mounted Architecture

To resolve the strain issues, Prototype 2 adopted a modular six-piece enclosure architecture. This design utilized a bottom-mounted internal layout, ensuring the Raspberry Pi, ADS1256, and AFE were safely anchored to a static base plate independent of the access panels.

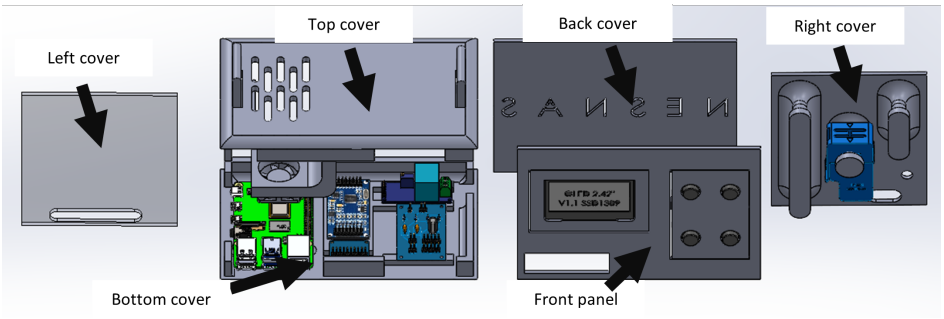


Figure A.4: CAD rendering of Prototype 2, demonstrating the modular six-piece architecture designed to eliminate wiring strain.

Extensive CAD work was also dedicated to the exterior interface panels, ensuring proper mounting tolerances for the OLED display, tactile buttons, and sensor input jacks. A specialized 2D DXF profile (Figure A.6) was generated to CNC-machine a transparent acrylic diagnostic window, completing the final mechanical enclosure utilized in the thesis.

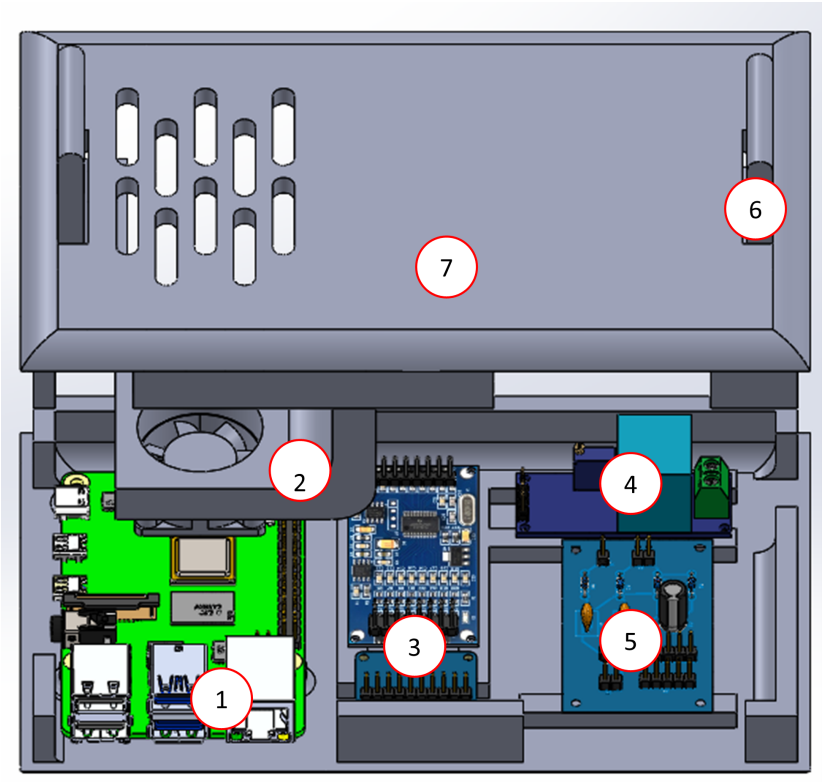


Figure A.5: Internal CAD topology of Prototype 2. Components are anchored safely: Raspberry Pi 4 (1), active cooling fan (2), ADS1256 (3), ZMPT101B (4), AFE board (5), chassis handles (6), and vented top cap (7).



Figure A.6: 2D DXF profile generated for the transparent diagnostic window.

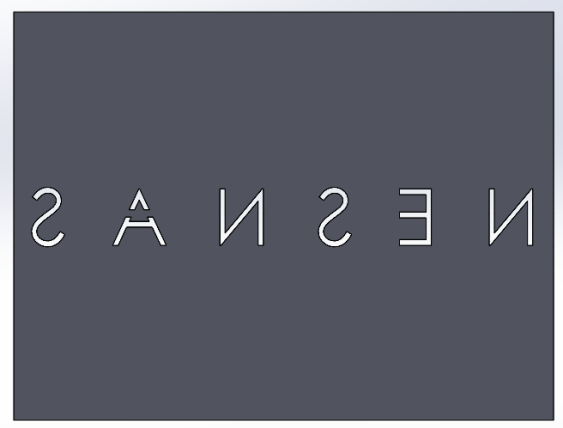


Figure A.7: CAD rendering of the rear chassis cap featuring the integrated NESNAS branding.

Appendix B

Software Architecture and Execution Logic

To maintain high-frequency metrology without being bottlenecked by the Python Global Interpreter Lock (GIL), the "Nesnas" Edge-AI software is strictly bifurcated. This appendix details the execution logic of the two primary threads: the high-priority hardware polling daemon and the localized Machine Learning inference engine.

B.1 The `NesnasCoreEngine`: DSP Pipeline

The core acquisition thread operates as an isolated loop. Its sole operational directive is to monitor the ADS1256 hardware interrupts, extract the 24-bit analog payloads, and perform the rigorous Digital Signal Processing (DSP) required to build the mathematical feature vector.

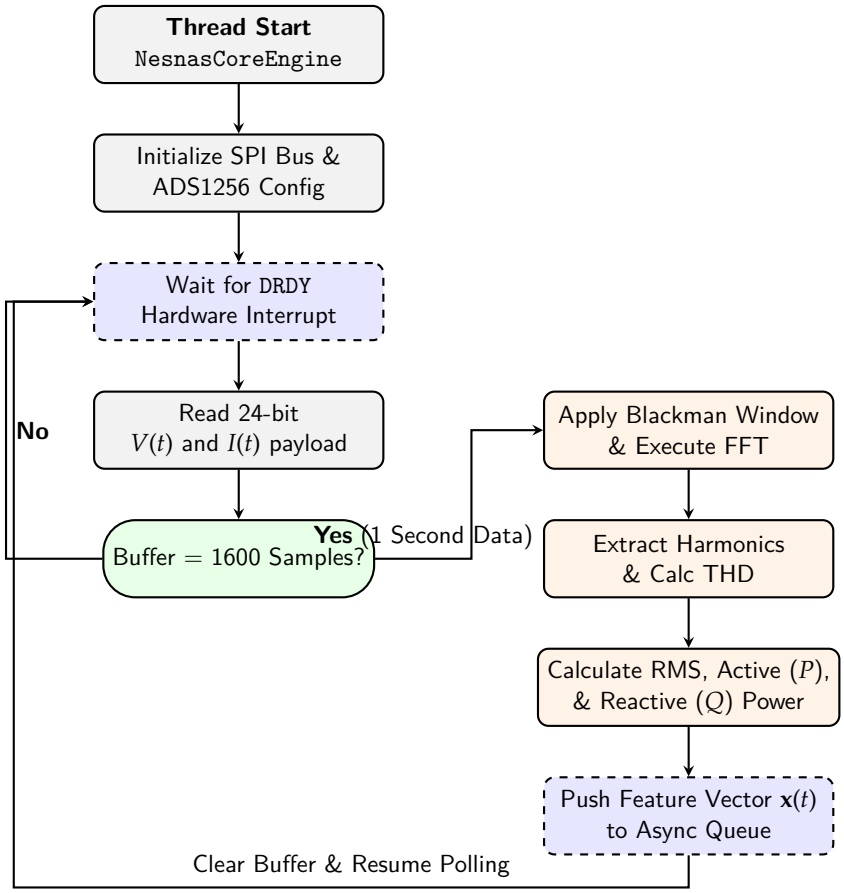


Figure B.1: Execution logic for the high-priority hardware polling and DSP thread. This loop strictly avoids blocking I/O to maintain the 1800 SPS empirical limit.

B.2 The Temporal Sliding-Window and Inference FSM

Operating completely asynchronously from the hardware loop, the Main Thread governs the localized AI logic. It consumes the mathematical feature vectors generated by the `NesnasCoreEngine`, buffers them into the 3-second Temporal Sliding-Window, and executes the serialized Random Forest classifier.

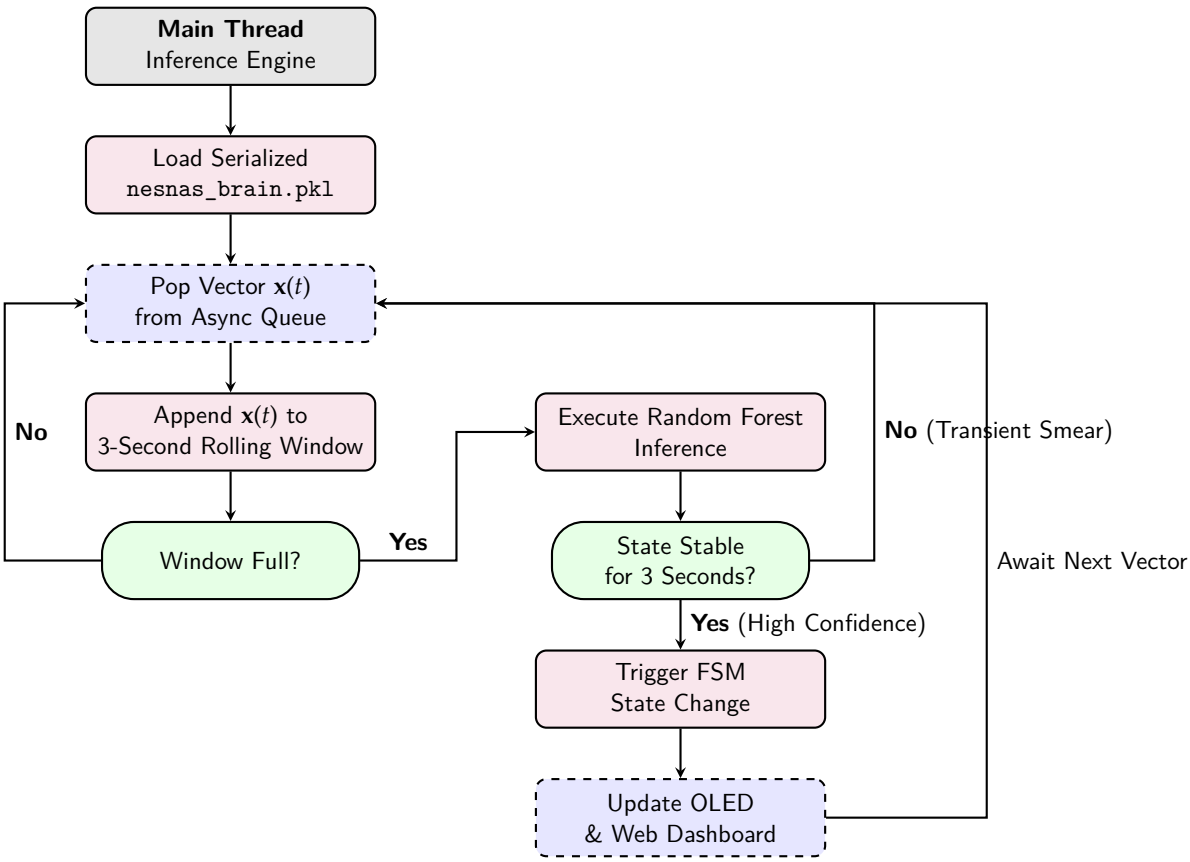


Figure B.2: Execution logic for the Inference Finite State Machine. The 3-second sliding window prevents the classifier from triggering false positives during physical appliance actuation.

Bibliography

[1] Leo Breiman. Random forests. Machine Learning journal, Volume 45, pages 5–32 2001.

[2] Asad Ali et al. Design of a smart energy meter using sct-013 and zmpt101b sensors for iot applications. *International Journal of Advanced Computer Science and Applications*, 10(9), 2019.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] David Cruz-Rangel, Carlos Ocampo-Martinez, and Javier Diaz-Rozo. Online non-intrusive load monitoring: A review. *Energy Nexus*, 17:100348, 2025.

[5] Jason A Donenfeld. Wireguard: Next generation kernel network tunnel. In *NDSS*, 2017.

[6] Pietro Ducange, Francesco Marcelloni, and Mauro Antonelli. A novel approach based on finite-state machines with fuzzy transitions for nonintrusive home appliance monitoring. *IEEE Transactions on Industrial Informatics*, 10(2):1185–1197, 2014.

[7] Electrical Engineering Stack Exchange. Positive part of a stereo jack. `https://electronics.stackexchange.com/questions/202061/positive-part-of-a-stereo-jack`.

[8] Electrocredible. Interfacing ZMPT101B ac voltage sensor with microcontrollers, 2023.

[9] George W Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.

[10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.

[11] HotCore Info. Raspberry Pi GPIO pinout — raspberry Pi 4 GPIO pinout.

[12] D. Jorde, M. Kahl, and H.-A. Jacobsen. Meed: An unsupervised multi-environment event detector for non-intrusive load monitoring. In *IEEE SmartGridComm*, 2019.

[13] Jack Kelly and William Knottenbelt. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pages 55–64, 2015.

[14] Christoph Klemenjak and Peter Goldsborough. Non-intrusive load monitoring: A review and outlook. *arXiv preprint arXiv:1610.01191*, 2016.

[15] Frederic Leens. An introduction to i2c and spi protocols. *IEEE Instrumentation & Measurement Magazine*, 12(1):8–13, 2009.

[16] Lars Mauch and Bin Yang. A new approach for supervised power disaggregation by using a deep recurrent lstm network. In *GlobalSIP*, pages 63–67, 2015.

[17] Open Source Hardware. Standard generic electronic breakout modules, 2023. Standard product reference photography.

[18] OpenEnergyMonitor. Ct sensors - interfacing with an ADC, 2023.

[19] F. Pedregosa et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] P. A. Schirmer and I. Mporas. Low-frequency energy disaggregation based on active and reactive power signatures. In *EUSIPCO*, pages 1426–1430, 2021.

[21] Pascal A. Schirmer and Iosif Mporas. Non-intrusive load monitoring: A review. *IEEE Transactions on Smart Grid*, 14(1):769–784, 2023.

[22] Claude E Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[23] Solomon Systech. *SSD1309 128 x 64 Dot Matrix OLED/PLED Controller Datasheet*, 2008.

[24] S. M. Tabatabaei, S. Dick, and W. Xu. Toward non-intrusive load monitoring via multi-label classification. *IEEE Transactions on Smart Grid*, 8(1):26–40, 2017.

[25] Texas Instruments. *ADS1255, ADS1256: 24-Bit, 30kSPS Analog-To-Digital Converters*, 2013. SBAS288K.

[26] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

[27] YHDC. *SCT013 Series Split Core Current Transformer Datasheet*, 2020.

[28] M. Zeifman and K. Roth. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics*, 57(1):76–84, 2011.

[29] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[30] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838–16866, 2012.