

# A Feature Reduction Strategy For Enabling Lightweight Non-Intrusive Load Monitoring On Edge Devices

Enrico Tabanelli, Davide Brunelli and Luca Benini

Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi" - DEI

University of Bologna, Italy

email: {name.surname}@unibo.it

**Abstract**—Non-Intrusive Load Monitoring (NILM) implies disaggregating the power consumption of individual appliances from a single power measurement point. Recent approaches use a mix of low and high-frequency features, but real-time NILM on low-cost and resource-constrained smart meters is still challenging due to the computing effort needed for feature extraction and classification. In this paper, we present a thorough survey on low, mid, and high-frequency features for enabling the deployment of NILM algorithms on edge-devices. We compare four different supervised learning techniques on different use-cases. Moreover, we developed a novel Microcontroller (MCU) based *Smart Measurement Node* for collecting measurements, providing computational capabilities to perform NILM on-the-edge. Experimental results demonstrate that by selecting the proper features, a robust disaggregation model for real-time load monitoring is feasible on our MCU-based meter with an accuracy of 95.99%, relying on merely 9.4kB of memory requirements and 16K MACs operation.

## I. INTRODUCITON

Non-Intrusive Load Monitoring (NILM) enables the disaggregation of the overall household power consumption taken from a single measurement point into appliance-level information. Since its introduction in 1992 [1], NILM has become an extremely active field of research. Recent approaches focus on the exploitation of a rich set of electrical features. The fusion of low and high-frequency features has been demonstrated to bring tangible benefits in load disaggregation [2]. However, high-frequency multi-feature NILM techniques rely on significant computing capabilities. As a consequence, the state-of-the-art framework on NILM requires server-based systems. Figure 1 depicts the typical *Remote NILM* architecture. A local meter collects household measurements, while a capable server-side backend performs computationally demanding disaggregation and classification techniques. Wireless communication enables a two-way data stream between the two sides, with high bandwidth reserved for data uplink.

Recent studies demonstrated that real-time appliance-level information could lead to energy savings over 5-20% [3]. Server-based frameworks do not scale appropriately in terms of communication bandwidth and privacy. Furthermore, they raise significant privacy concerns. Hence, there is a critical need for effective load monitoring and analysis based on in-situ data processing.

Enabling load monitoring on edge-devices is challenging due to memory and computing power constraints. As a

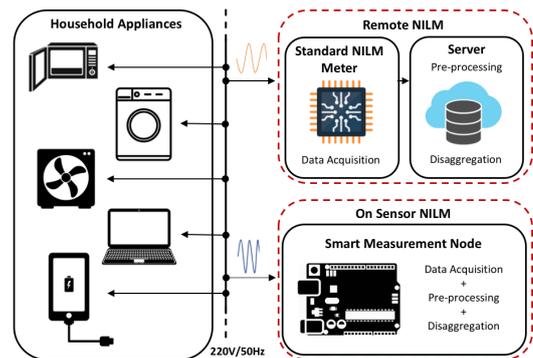


Fig. 1: Non-Intrusive Load Monitoring Architecture

consequence, the trade-off between algorithm complexity and accuracy needs to be explored to design a smart node that performs local, real-time NILM.

Low-cost commercial power meters are not well-suited for running advanced NILM algorithms because they do not offer sufficient computational capabilities, and often they cannot even sample current and voltage at a high enough frequency. To tackle the shortcomings of current power meters, we developed a novel *Smart Measurements Node*. As shown in Figure 1, we designed the device to cut out remote server processing. In this regard, we need to carefully assess the set of features that need to be extracted, satisfying the limitations of resource-constrained embedded processors. Furthermore, we need to also assess the memory demands of the disaggregation algorithm, since memory severely impacts cost.

The paper presents the following contributions:

- 1) We developed a low-power MCU-based data acquisition system, the *Smart Measurement Node*, capable of high-frequency current and voltage measurements. The on-board modules enable wireless data streaming as well as ultra-low-power computation.
- 2) We present an in-depth survey of electrical frequency features and an analysis of the trade-off between computing overhead and disaggregation accuracy. Our analysis shows that a limited set of low-frequency features is suitable for accurate at-the-edge monitoring with affordable computational effort.
- 3) We compare four memory-constrained supervised learning techniques on load classification scenarios, showing

that a Random Forest (RF) algorithm reaches an accuracy of 96.76% on a complex extended event-based scenario.

The paper is organized as follows. In Section II, we present related works, and we discuss the recent approaches for load disaggregation. Section III describes the hardware of the *Smart Measurement Node*. Section IV highlights the crucial stages of the NILM framework, including data acquisition, features extraction, and disaggregation. Experimental results of our exploration of feature extraction and disaggregation algorithms are reported in section V. Concluding remarks and plans for future works complete this paper.

## II. RELATED WORK

Non-Intrusive Load Monitoring (NILM) is the task of disaggregating the power consumption of individual appliances from a single current and voltage measurement point, usually at the main power inlet of a household. The NILM research field is vast and includes several different areas. A significant preliminary partition is between supervised and unsupervised learning. As stated by Klemenjak [4], the distinctive factor is whether or not ground-truth data is available for the training phase. In this work, we manually labeled data and examined only supervised algorithms.

The seminal paper on load disaggregation is by Hart [1]. He introduced different NILM scenarios and implemented a model-based approach to accurately decompose the aggregate load into individual components using low-frequency features. His approach presents fair accuracy when using ordinary ON/OFF appliances but fails when applied to more sophisticated appliances, such as Finite State Machine appliances (FSM) or Continuously Variable Devices (CVD). Many researchers tried to disaggregate the mains power using real power as a single feature [5]. Unfortunately, this approach does not discriminate loads with similar power consumption characteristics. Others tried to overcome the limitations of power-based methods analyzing current and voltage waveforms [6]. The method proved to be successful, excluding CVD loads and simultaneous appliances activation.

Along with the growing interest in Machine Learning (ML), ML-based NILM method gained popularity. These approaches differentiate in the choice of input features and classification algorithm [7]. Many algorithms use data collected at sampling rates around one Hz, like J. Kelly [8] that introduced the first application of Neural Networks (NN) to NILM. Other methods based on the use of Fourier series analysis have gained traction for classifying non-linear and complex dynamic loads [9]. In this regard, a recent approach samples at one MHz, and extracts EMI features in the frequency domain to classify using a kNN algorithm [10].

While the most significant advantage of the low-frequency approach is its applicability in low-cost smart meters, the higher frequency approach can recognize very hardly distinguishable loads. Since both types have their shortcomings, the method explored in this paper combines low and high-

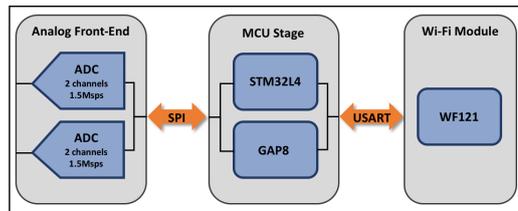


Fig. 2: Schematic overview of the *Smart Measurement Node*

frequency features following the proposal of T. Bernard et al. [2] for a single-channel blind source separation problem.

## III. SYSTEM ARCHITECTURE

The *Smart Measurement Node* integrates two microcontroller units (MCU), as shown in Figure 2. One is the STM32F4, a 32bit STMicroelectronics MCU based on the ARM Cortex-M4 core with 128kBytes of FLASH memory and 32kBytes of RAM. The core runs up to 100MHz and allows a computing power of 1.25DMIPS/MHz. In run mode, dynamic power scaling enables the current consumption to be as low as 89  $\mu\text{A}/\text{MHz}$ . The second MCU is GAP8, a 32bit ultra-low-power IoT-edge computing engine. The System-on-Chip (SoC) features a Fabric Controller (FC) core coupled with an 8-core Cluster. All cores support the same extended RISC-V instruction set architecture (ISA) that enables DSP-centric operations, such as Single Instruction Multiple Data (SIMD) instructions. GAP8 does not include data caches but features a 512kBytes of L2 memory accessible by all cores and two smaller L1 memories: 16kBytes for the Fabric Controller (FC) and 64kBytes as multi-banked shared scratchpad memory for the Cluster. Moreover, the SoC integrates two instruction caches: 4kBytes for the FC and 16kBytes for the Cluster. The extremely energy-efficient design enables a deep-sleep power of 3.6  $\mu\text{W}$  and allows up to 10 GMAC/s (90 MHz, 1.0V) at the energy efficiency of 600 GMAC/s/W within a worst-case power envelope of 75 mW. In this work, the STM32F4 is in charge of measurement settings, data acquisition, and streaming it to a server. GAP8 will be used in the future to run the NILM algorithm in-situ.

The node features an analog front-end consisting of two instances of the LTC1407A, a dual-channel ADC from Linear Technology capable of sampling rates up to 1.5Msps while recording simultaneously, and a resolution up to 14bit with 16384 discrete digital values. As a result, the 0V to 2.5V unipolar full-scale input range leads to a voltage resolution of 152  $\mu\text{V}$ . The 80dB Common Mode Rejection Ratio (CMMR) at 100kHz allows eliminating efficiently common-mode noise by measuring signals differentially from the source. On the other hand, the 74 dB Signal-to-Noise Ratio (SNR) at 100Hz highlights the low-noise performance. The typical power consumption of 14mW contributes to the overall energy efficiency of the node. Furthermore, the analog stage provides an Isolated Interface, which contains a voltage divider and a Shunt resistor to measure voltage and current, and a Non-Isolated Interface, with the possibility to use Rogowski Coils and Hall-Effect Sensors. In this paper,

we used the Isolated Interface because it allows for direct and simultaneous sampling of current and voltage.

The system also integrates the WF121 WiFi module by Bluegiga Technologies. The device features a 2.4GHz 802.11 b/g/n radio and a 32bit MCU, which provides low-level programming drivers and an API for typical use cases. To test the Wi-Fi bandwidth, we send 256-byte-sized packets to a server awaiting the receiving end. The measured net bandwidth on a total transmission size of 2.56MBytes is 800kbps, which translates into an upload sample rate of 57ksps (28.5ksps, respectively) with 14bit of sample resolution.

The firmware running on our *Smart Measurement Node* consists of different stages. A preliminary phase establishes a Wi-Fi connection to the client-server as well as switches the Wi-Fi module in streaming mode. An advanced control timer sets a sampling rate of 20kHz by enabling the ADC to start acquiring current and voltage measurements. Considering STM low-level drivers work only with multiples of bytes, we store two 14bit samples in 4-Byte-Arrays. The four unused bits allow marking buffer overflows. The Serial Peripheral Interface (SPI) streams data operating at a frequency of 16Mbps, while on the active MCU, the Direct Memory Access (DMA) handles the reception. To reduce the noise induced by the Wi-Fi module, we average two consecutive measurements, which results in an actual sampling rate of 10kHz. A 512Bytes Ping-Pong buffer helps boosting the overall throughput overlapping I/O task and data processing. When one buffer is full, we transmit its content via Universal Synchronous-Asynchronous Receiver/Transmitter (USART) to the Wi-Fi module, therefore readily streamed to the receiving client. A baud-rate of 1Mbps is enough to enable continuous dual-channel 14bit measurements. On the server-side, we extract features and process data by using Scikit-learn ML libraries.

#### IV. NILM FRAMEWORK

The standard NILM framework consists of three stages. A data acquisition system collects electrical signals as voltage and current. In the feature extraction stage, electrical features, such as real power and harmonics, are extracted. Finally, the disaggregation algorithm processes the features leading to appliance-level information.

##### A. Data Acquisition

To analyze typical NILM scenarios, we recorded household appliances that appear in publicly available NILM Dataset, such as BLUED [11]. Our dataset, openly accessible at [12], includes ten different devices from three categories described in the literature [1]:

- ON/OFF appliances: Two-states loads such as the light bulb, the electric coffee machine, and the fan. Even if the fan has three different power states related to fan speed, we considered them belonging to the same group.
- FSM appliances: Multi-state devices with a finite number of working states. The microwave oven represents a

FSM device because its magnetron repeatedly turns on and off with a transition state in between.

- CVD appliances: Their power consumption changes continuously depending on the battery charge level and current workload with no consistent step change. The monitor, the laptop, the smartphone charger, the headphones, the power-bank, and the waveform generator are CVD devices belonging to our dataset.

To enable load disaggregation on event-free and event-based NILM scenarios, we divided the dataset into sections, each for a distinct training and testing phase.

- (a) Single Appliances without Switching Events: A Dataset composed of each appliance individually recorded at a steady state. The graph in Figure 3.a shows the typical sine-wave electric current envelope of a linear load, the fan. This class of acquisitions gives us a clear overview of devices' electric current waveform and is highly practical for event-free classification.
- (b) Single Appliances with Switching Events: Recordings consisting of each appliance, individually collected, switched on in the middle of the trace. In Figure 3.b, we observe the electric coffee machine current when switching from OFF to ON in the first part of the recording. These measurements target both event-free and event-based load classification.
- (c) Multi-Appliances with/without Superposition: A realistic scenario built by a restricted set of appliances with two different switching patterns. The first one includes switching on/off appliances without superposition. In the second one, each appliance is individually switched on, then switched off, keeping the same order. Figure 3.c shows the current superposition of several appliances. From the nearly sinusoidal waveform, it becomes clear the presence of linear loads, such as the light bulb and the monitor, while the recurring peaks are due to Switched-Mode Power Supply (SMPS) appliances, such as the power-bank. These measurements are suitable to validate both event-free and event-based scenarios.
- (d) Extended Single Appliances: Data collected to test the chance of adopting EMI features with more complex appliances. We recorded a late 2011 MacBook Pro with an Intel I5 core running from one to four threads. To effectively stress the laptop, we issued a command from the terminal, which pushes the CPU until its limit. We also collected the laptop when running a 4K video playback to have a different workload comparison. We recognize in Figure 3.d the ordinary current envelope of a SMPS as well as CVD device, such as the MacBook Pro running four threads.

##### B. Feature Extraction

In this work, we examined the deployment of different feature vectors composed of low and high electrical frequency features. By multiplying current and voltage samples, we obtain the instantaneous power. Within a time-frame of 100ms long, accordingly 1000 instantaneous power values,

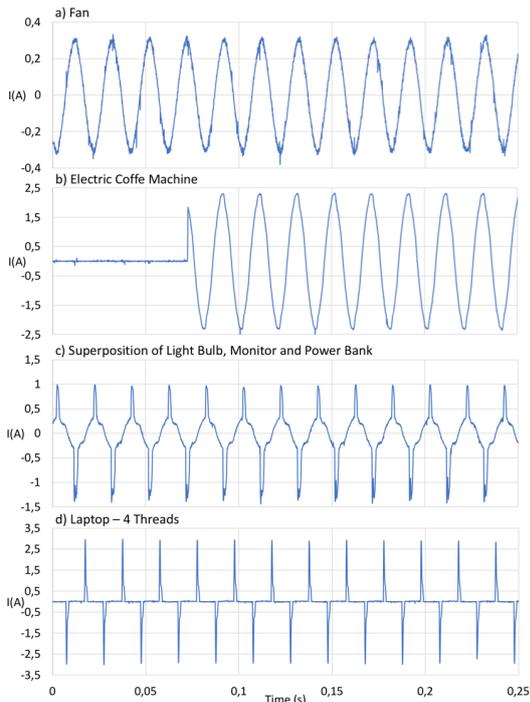


Fig. 3: Electrical Current Acquisition in 4 Different Scenarios

we average them and then compute the active, nonactive, and apparent power. To determine higher frequency features, we calculate the Discrete Fourier Transform (DFT) of electric current samples over the same time-frame. Because odd harmonic currents constitute efficient features for load disaggregation, we extract them, resulting in 100 values per time-frame. The sampling frequency of 10kHz results in a maximum measurable frequency of 5kHz at a resolution of 10Hz. Low-frequency features form the first part of the generated feature vector, while high-frequency features the second part.

The feature extraction trade-off between frequency components and computing resources needed for obtaining them restricts NILM implementation severely. While the number of low-frequency features makes the computing task lightweight for resource-constrained devices, DFT has a computational complexity of  $O(N \times \log(N))$ , where  $N$  is the number of samples. The required computing resources makes an on-the-edge feature extraction challenging when extending the frequency range. To evaluate the opportunity of reducing the feature vector size for enabling lightweight NILM on resource-constrained and low-power MCUs, we considered four different feature vectors:

- (I) Low-frequency features;
- (II) Odd electric current harmonics up to the  $10^{th}$ .
- (III) Low-frequency features and odd electric current harmonics up to the  $10^{th}$ ;
- (IV) Low-frequency features and odd electric current harmonics up to the  $50^{th}$ ;

In table I, we reported the computational complexity in terms of Multiply-Accumulation (MAC) operations required to extract the feature vector in each of the above cases. Push-

ing the extraction process toward high-frequency features clearly demands to increase the computing effort.

TABLE I: Feature Vector Extraction Computing Effort

Feature Vector	MACs
Low-freq.	3K
10 High-freq.	4.6K
Low-freq. + 10 High-freq.	7.6K
Low-freq. + 50 High-freq.	7.7K

### C. Disaggregation Algorithms

In the disaggregation stage, supervised learning techniques process the extracted features breaking down the overall load profile into appliance-level information. To enable NILM on-the-edge in the future, we focused on the deployment of memory-efficient and low computational algorithms. Accordingly, we evaluated the performance of a k-Nearest Neighbor (kNN), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and Random Forest (RF) classifiers. Below, we briefly introduce the algorithms mentioned above.

- kNN: A non-parametric algorithm based on feature similarity. It can classify an unknown input from the test set by its k-nearest neighbor learned during the training phase. Its simplicity and strength as regards the research space, along with state-of-the-art achievements obtained by Bernard [2], make this model interesting for comparisons.
- SVM: A supervised machine learning method capable of performing multi-class classification. It relies on finding a set of hyper-planes in high dimensional space to classify instances. The algorithm already proved its potential in many load classification scenarios [13]. As a consequence, we decided to include it in our experiment.
- MLP: A feed-forward Artificial Neural Network (ANN) consisting of at least three fully connected layers. Its capability of learning non-linear models and its flexibility, together with the results achieved in NILM by Kelly [8], make it an attractive option to evaluate for our approach.
- RF: An easy-to-use classifier consisting of several multiple decision trees created at training time. Each tree provides a class prediction for an input object, and then the model aggregates the votes to decide the final class. The high accuracy achieved in different classification scenarios makes Random Forest models challenging for load disaggregation.

In Table II, we schematize the computational complexity and the memory footprint of each algorithm. The memory demand consists of the number of parameters stored at inference time into the edge-device. At the same time, the asymptotic complexity refers to the computing resources needed for running the algorithm. This latter accounts for the number of required MAC operations during a single feed-forward pass, except for the RF classifier where comparisons (CMP) are the most executed operations. The table clearly outlines the linking between resources and feature vector

TABLE II: Algorithm Computational Complexity and Memory Footprint

Model	Computational Complexity	Memory
SVM	$O(N_{features} \times N_{sv})$	$N_{sv}$
RF	$O(N_{features} \times N_{trees})$	$N_{threshold} \times N_{trees}$
KNN	$O(N_{features} \times Dataset_{size})$	$N_{features} \times Dataset_{size}$
MLP	$O(\sum_{i=0}^{Layer-1} In_i \times Out_i)$	$N_{weights}$

dimension, further highlighting the need to reduce the feature vector size to reduce the overall computing effort.

## V. EVALUATION

We detail here the results of our analysis. The evaluation covers the three classification scenarios and the four machine learning approaches mentioned in the previous section. We divided the procedure in (i) a preliminary hyperparameter search, to find the most performing algorithm, and (ii) an evaluation stage to assess the model performance. The first stage considers only accuracy as a metric and consists of:

- 1) Grid Search to tune hyperparameters;
- 2) 10-Fold Cross-Validation to further test parameters;
- 3) Comparison to find the most accurate algorithm.

The second one includes:

- 1) Performance evaluation on the dataset split in 2/3 for the training set and 1/3 for the testing;
- 2) Performance evaluation on a dataset from a different scenario.

In the latter stage, we used precision, recall, and F1 score, as well as accuracy.

### A. Event-Free Load Detection

We trained the algorithms considering a dataset composed only by instances from the scenario (a), meaning without switching events. The model is fed with the extracted features and is used to predict appliances in the recordings. The hyperparameter optimization procedure (i) resulted in a linear kernel SVM with a feature vector consisting of power measurements as well as electric current harmonics up to the 10<sup>th</sup> harmonics. The generated model needs 293 Support Vectors  $N_{sv}$  to identify the optimal separating hyperplane to maximizes the margin of the training data, while 3809 MAC operations are required to perform the algorithm. The other models provide good results, as well. In table IV, we

TABLE IV: Event-Free Performance with III Feature Vector over 10-Fold Cross-Validation

Model	Parameters	Complexity	Memory (kB)	Mean Accuracy (%)
SVM	$N_{sv} = 293$	3809 MAC	9.4	99.96
RF	$N_{threshold} = 50, N_{trees} = 10$	500 CMP	16	99.90
KNN		$Dataset_{size} = 331k$	4.3M MAC	10600
MLP	$N_{weights} = 6650$	6650 MAC	213	99.76

TABLE V: Event-Free Performance with II Feature Vector over 10-Fold Cross-Validation

Model	Parameters	Complexity	Memory (kB)	Mean Accuracy (%)
SVM	$N_{sv} = 285$	2.85k MAC	9.12	99.87
RF	$N_{threshold} = 10, N_{trees} = 1000$	10k CMP	32	99.33
KNN		$Dataset_{size} = 331k$	3.3M MAC	10600
MLP	$N_{weights} = 1.6k$	1.6k MAC	51.2	99.85

show algorithms accuracy together with parameters, MAC operations, and memory footprint.

In the evaluation stage (ii), we trained the SVM on a dataset extended to measurements without superposition from the scenario (c). From results shown in Table III, we notice that, even though a fraction of accuracy is lost, the model still has remarkable accuracy.

TABLE III: SVM Event-Free Performance on Seen/Unseen Data

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Seen	99.90	99.91	99.90	99.90
Unseen	95.99	98.55	95.99	97.13

Excluding power measurements from the feature vector, still enables load classification with exceptional results. As reported in Table V, using only electric current harmonics up to the 10<sup>th</sup> led to a low memory-footprint and computationally lightweight SVM with 99.87% of accuracy.

### B. Event-Based Load Detection

To detect appliances in an event-based context, we designed an algorithm described by the flow chart in Figure 4. Below, we describe the system behavior:

- 1) Voltage and current acquisition at a sampling rate of 10kHz;
- 2) Active power calculation over a time-frame of 100ms;
- 3) Triggering a switching event when a variation in the active power exceeds a certain threshold;
- 4) Estimation of the differential feature vector  $F_{diff}$  after the event detection trigger;
- 5) Appending the differential feature vector to the dataset;
- 6) Training of the model on the new dataset.

The crucial step concerns the estimation of the differential feature vector, which combines features from different instants regarding the switching event. We extract features immediately preceding  $F_{n-1}$  and following  $F_{n+1}$  the event, as well as 10 and 20 samples previously,  $F_{n-10}$  and  $F_{n-20}$ , and next,  $F_{n+10}$  and  $F_{n+20}$ , to the event. We average the feature vectors before and after the event resulting in two intermediate vectors, whose subtraction leads to the differential feature vector.

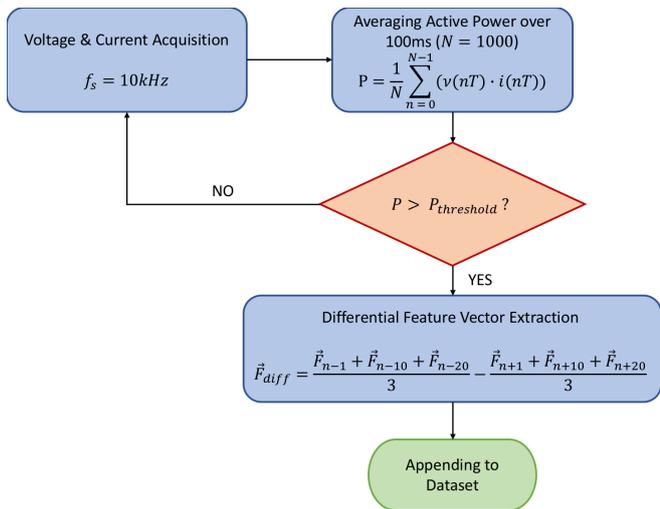


Fig. 4: Event-Based Detection Flow Chart

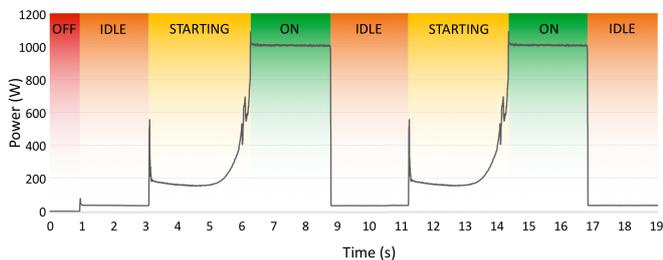


Fig. 5: Microwave oven Power Intake

The choice of the power threshold is of paramount concern. The issue at stake is a trade-off between a drop in the threshold, to raise low-power devices detection, and the increase of false positive, especially for high-energy appliances where power fluctuations are more substantial. To achieve the highest-performance on event detection, we computed the likelihood detection when varying the power threshold between 0W to 10W. The experiment resulted in 5W as the most effective threshold, leading to a likelihood of 99.4% to detect switching events.

Several appliances present oscillating transients when changing the operating states. Because transitioning shows a significant power-level shift, we temporarily disable the detection scheme setting a hysteresis time of 5s to avoid false positive. Accordingly, we observed from the results of the likelihood test that our detection system fails mostly on appliances with several adjacent changing of power states. As revealed from the microwave oven average active power intake in Figure 5, the magnetron works through cycles that alternate between fully-on and fully-off, thus altering the large scale duty-scale. In between the IDLE and ON phase, a STARTING transitional state allows the magnetron to reach the fully-on activity. The FSM easily enables the OFF/IDLE and STARTING/ON event detection. At the same time, the short elapsed time in between consecutive states does not allow to detect the following IDLE/STARTING and ON/IDLE event.

To assess performance, we restricted the deployed appliances to a set formed by the power-bank, the light bulb,

the monitor, the electric coffee machine, and the fan at minimum speed. We built the dataset with instances from the scenario (b), namely with switching events. The in-depth parameter search (i) led to a RF classifier as the most accurate algorithm. Power measurements, as well as current harmonics up to the 10<sup>th</sup> harmonics, turned out to be the most suitable feature vector. The accuracy obtained in the 10-Fold Cross-Validation procedure is 98.89%. The classifier consists of  $N_{trees}=100$  estimators and  $N_{leafnodes}=10$  leaf nodes, thus leading to a memory requirement of 32kB to store threshold values and to a worst-case computing effort of 10K comparisons (CMPs).

TABLE VI: RF Event-Based Performance

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Seen	98.35	98.55	98.35	98.45
Unseen	97.67	97.89	97.67	97.66

In the evaluation stage ii), we added unseen instances with superposition from the scenario (c). We reported in Table VI the performance scores on both seen and unseen data. Although there is a low decrease in performance, the model still performs well, achieving an accuracy of 97.67%.

The results demonstrate that our approach provides competitive accuracy versus state-of-the-art event-based approaches and, more importantly, enables lightweight NILM suitable for limited computing capabilities and resource-constrained embedded systems. In fact, Bernard's kNN-based approach [3] on event-based scenario achieves an F1 score of 93%, demanding hundreds MB to store the 3-day dataset and considerable computing resources for pre-processing and disaggregating. Our RF event-based approach reaches an F1 score of 97.66%, requiring only 13k MAC operations for the reduced feature vector extraction and 10K comparisons (CMP) for each inference, while the memory footprint is limited to 32kB.

### C. Extended Event-Free Load Detection

The last load classification scenario reflects a more complex context, developed to explore the capability of our approach to classifying even subtle differences in current and voltage traces, such as those created by a change in the workload of a laptop. The test dataset consists of instances from the scenario (a) and (d). As a result, we added classes related to laptop states.

The comprehensive hyperparameters exploration (i) resulted in pure power measurements as the most efficient feature vector. kNN classifier enables load detection with remarkable accuracy at the expense of 3MB memory demand to store dataset instances. Large memory footprints such as this are impractical for low-cost resource-constrained MCUs. As a result, we focused on a more memory-lightweight technique suitable for edge computing limitations. By demanding only 5K comparisons (CMP) and 160kB of memory requirement, the RF classifier achieved an accuracy of 92.14% on the 10-Fold Cross-Validation. In Table VII, we illustrate the accuracy achieved by the two algorithms together with parameters, MAC operations, and memory footprint.

TABLE VII: Extended Event-Free Performance over 10-Fold Cross-Validation

Model	Parameters	Complexity	Memory (kB)	Mean Accuracy (%)
RF	$N_{threshold} = 50, N_{trees} = 100$	5k CMP	160	92.14
KNN	$Dataset_{size} = 32k$	96k MAC	3072	97.79

True Label	Headphones	Laptop	Lightbulb	MACvideo	MACthread1	MACthreads2	MACthreads3	MACthreads4
Headphones	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Laptop	0.0	1.0	0.0	0.16	0.04	0.01	0.0	0.0
Lightbulb	0.0	0.0	0.79	0.0	0.0	0.0	0.0	0.0
MACvideo	0.0	0.1	0.0	0.89	0.0	0.0	0.0	0.0
MACthread1	0.0	0.0	0.0	0.0	0.93	0.02	0.01	0.02
MACthreads2	0.0	0.0	0.0	0.0	0.02	0.82	0.08	0.08
MACthreads3	0.0	0.0	0.0	0.0	0.03	0.1	0.8	0.07
MACthreads4	0.0	0.0	0.0	0.0	0.02	0.05	0.06	0.86

Fig. 6: Extended Event-Free Confusion Matrix

The confusion matrix reported in Figure 6 gives some relevant insights. To increase the readability, we cropped the central section of the original one. The model can classify laptop states against other appliances correctly. Nevertheless, there are some misclassifications. It happens mostly between MACthreads2, MACthreads3, and MACthreads4, while MACthread1 seems to be well recognized. We remark that the accuracy of each appliance is always above 79%, which is a very promising result.

TABLE VIII: RF Extended Event-Free Performance

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Seen	91.92	92.08	91.92	91.92
Useen	96.76	99.25	96.76	97.91

To check against overfitting, we extended the dataset to measurements without the superposition of scenario (c). The results, shown in Table VIII, confirm that the model still enables load classification on unknown instances achieving an accuracy of 96.76%.

## VI. CONCLUSION

To fully exploit NILM potentiality, real-time feedback on appliance power consumption is crucial. State-of-the-art techniques on NILM rely on large memory-footprint and high-power computing resources, but modern smart meters are resource-constrained and cannot afford those workloads.

To overcome this computational and storage bottleneck, we developed a novel flexible and low-power *Smart Measurement Node*. The device enables voltage and current acquisition up to a sampling rate of 1.5MHz and a Wi-Fi bandwidth of 800kbps, meaning 57ksps with 14bit of sampling resolution. We designed the system with two larger

power computing on-board MCUs to allow NILM at-the-edge. To reduce feature extraction computing requirements as well as disaggregation memory footprint, we provided an in-depth analysis of low and higher electrical frequency features together with a survey of the trade-off between NILM accuracy and computing overhead. We tested four different feature vectors on four different supervised machine learning techniques. The experiments led us to conclude that a reduction of feature vector size, without compromising accuracy is possible. A SVM classifier requiring 9.4kB of 32bit support vectors and 3.9k MAC operations per inference reaches an accuracy of 96.74% on the event-free dataset. As for event-base classification, we achieve accuracy comparable with SOA, while only requiring requiring 13K MACs for feature vector extraction and 10K comparisons (CMP) per inference with just 32kB memory-footprint. This results demonstrates feasibility of NILM on edge devices. A more extensive feature vector allows for detecting subtler events. We enable laptop workload classification with accuracy always above 79% per each thread class, requiring just 160kB of storage.

The future work will target a fully on-the-edge NILM implementation. The focus will be on porting the full NILM processing on the two MCUs available on our *Smart Measurement Node* enabling live disaggregation. The presence of the single-core (STM32F4) and multicore (GAP8) will allow a parallel acceleration of the NILM algorithm.

## VII. ACKNOWLEDGMENTS

The research is funded by the EU H2020-ECSEL project Arrowhead Tools (g.a. 826452).

## REFERENCES

- [1] G. Hart, "Nonintrusive appliance load monitoring," Proceedings of the IEEE, vol. 80, no. 12, pp. 1870-1891, 1992.
- [2] T. Bernard, M. Verbunt, G. vom Bogel, and T. Wellmann, "Non-Intrusive Load Monitoring (NILM): Unsupervised Machine Learning and Feature Fusion : Energy Management for Private and Industrial Applications," in 2018 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE 2018): 29 May-1 June, 2018, Kajang, Malaysia, Sg. Long, Cheras, Kajang, Malaysia, 2018, pp. 174-180.
- [3] S. D by, "The effectiveness of feedback on energy consumption," Environmental Change Institute, University of Oxford, Oxford, UK, Tech. Rep., 2006.
- [4] Christoph Klemenjak and Peter Goldsborough. Non-intrusive load monitoring: A review and outlook. arXiv preprint arXiv:1610.01191, 2016.
- [5] Marceau, M.L.; Zmeureanu, R. Nonintrusive load disaggregation computer program to estimate the energy consumption of major end uses in residential buildings. Energ. Convers. Manag. 2000, 41, 1389-1403.
- [6] Figueiredo, M.; de Almeida, A.; Ribeiro, B. An experimental study on electrical signature identification of Non-Intrusive Load Monitoring (NILM) systems. In Adaptive and Natural Computing Algorithms; Dobnikar, A., Lotric, U., Ter, B., Eds.; Springer: Berlin, Germany, 2011; Volume 6594, pp. 31-40.
- [7] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," IEEE Trans. Consumer Electron., vol. 57, no. 1, pp. 76-84, 2011.

- [8] J. Kelly et al., "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. ACM, 2015, pp. 55–64.
- [9] Li, J.; West, S.; Platt, G. Power Decomposition Based on SVM Regression. In Proceedings of International Conference on Modelling, Identification Control, Wuhan, China, 24–26 June 2012; pp. 1195–1199.
- [10] S. Gupta, M. S. Reynolds, and S. N. Patel, "ElectriSense: Single-point Sensing Using EMI for Electrical Event Detection and Classification in the Home," in Proceedings of the 12th ACM International Conference on Ubiquitous Computing, 2010, pp. 139–148.
- [11] Anderson, K.; Ocneanu, A.; Benitez, D.; Carlson, D.; Rowe, A.; Berges, M. BLUEED: A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research. In Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability, Beijing, China, 12–16 August 2012.
- [12] <https://github.com/EnricoTabanelli/NILM-Dataset>
- [13] Lin, Yu-Hsiu, Tsai, Men-Shen (2011). 'Applications of Hierarchical Support Vector Machines for Identifying Load Operation in Nonintrusive Load Monitoring Systems'. In: 9th World Congress on Intelligent Control and Automation. WCICA. IEEE, pages 688–693. doi: 10.1109/WCICA.2011.5970603 (cited on pages 135, 143).