eNILM: Dynamic Device Integration in Non-Intrusive Load Monitoring

Justus Breyer
and Kilian Plachetka
Communication and Distributed Systems
RWTH Aachen University, Germany
Email: breyer@comsys.rwth-aachen.de

Muhammad Hamad Alizai
LUMS
Lahore, Pakistan
Email: hamad.alizai@lums.edu.pk

Klaus Wehrle
Communication and Distributed Systems
RWTH Aachen University
Aachen, Germany
Email: wehrle@comsys.rwth-aachen.de

***Abstract*—We introduce eNILM, an *extensible* Non-Intrusive Load Monitoring (NILM) framework capable of simultaneously integrating multiple new devices at runtime. Leveraging a mixture of experts using density-based outlier detectors, such as Local Outlier Factor and Isolation Forest, and clustering algorithms such as DBSCAN and HDBSCAN, eNILM dynamically adapts to changes in household device composition. The framework eliminates the need for retraining existing classifiers, enhancing its scalability and flexibility. We conducted extensive evaluations on the public datasets FIRED and DARCK, demonstrating that eNILM achieves over 85% accuracy in filtering out unknown devices and over 91% accuracy in clustering unknown devices on average. These results highlight eNILM's potential to substantially improve the practicality and reliability of NILM systems.**

I. Introduction

Motivation. The escalating impact of climate change has underscored the need to reduce greenhouse gas emissions [1]. To provide home-owners with insights on their energy consumption patterns in order to motivate behavioral changes, Non-Intrusive Load Monitoring (NILM) [2] analyzes total household electricity consumption measured at a single point using time series analysis and machine learning (ML) techniques. Compared to installing separate meters for each appliance, the maintenance and networking effort is substantially lower.

NILM Pipeline. NILM systems are commonly categorized as either eventless or event-based. eNILM falls into the latter category. In event-based NILM, aggregate power data is analyzed to detect device state changes, such as an appliance turning on, which usually appear as sudden shifts in power consumption. A range of methods has been proposed for this step, from expert heuristics to probabilistic models and matched filters [3]. According to the Switch Continuity Principle [2], each event can be attributed to a single device state change. Once detected, events are represented by feature vectors extracted from a small region of interest around each event [4] and then classified using ML models [2]. Finally, device-level energy consumption is estimated from the classified events and per-state power estimates. The final step depends heavily on the correctness of classification, which is why much of the NILM literature has focused on improving this earlier stage.

Deployment Challenge. A significant limitation of NILM, however, is its dependence on specific models and training data. This reliance often confines NILM systems to a predefined range of devices, limiting their flexibility in residential settings where the set of appliances may change over time. For multi-class NILM systems, a critical challenge is the misclassification of unknown or untracked devices [3]. This issue leads to erroneous energy disaggregation and undermines system accuracy. In a brief investigation presented later in this paper, we show that the median classification accuracy of a static classifier can drop drastically, from 98.5% to 26.4%, when unknown devices are introduced. This sharp decline occurs because unknown events inevitably receive incorrect labels. To mitigate this issue, classifiers must be manually updated, creating a significant barrier to the practical deployment of NILM.

Prior Work. Most NILM research over the past three decades has focused on static scenarios with a fixed number of devices, aiming to improve the discrimination of device signatures [3]. In real deployments, however, the composition of household devices is dynamic. New devices can be introduced at any time, and previously unmonitored devices may generate signatures unknown to the model. Several studies have therefore addressed the problem of detecting unknown appliances [3], [5]–[8], however, without incorporating them into the NILM system.

Other work has enabled extensibility [9], [10], however, without automatic detection of new devices. Lan et al. [11] proposed active learning by incorporating a novelty detection branch into a neural network. Gillis and Morsi [12] used Wavelet analysis to automatically identify the appearance of a new device, but their approach is limited to a single unknown load. More broadly, current NILM systems still struggle to scale to the number of devices found in a household [13], making unmonitored loads a common practical scenario.

Our Approach. An alternative to manually updating ML models in NILM is to automate model adaptation, relying on the system's ability to assess its own performance on incoming events and trigger updates when necessary. However, this is challenging in residential environments, where new devices may appear at any time and the distinguishing features of future unknown devices cannot be specified in advance. Despite several promising efforts, existing approaches do not jointly provide automated detection of changing device composition and straightforward integration of multiple unknown devices.

We introduce eNILM, an extensible NILM framework capable of simultaneously integrating multiple new devices at

runtime. eNILM is built on density-based ML models that filter unknown devices and generate training data through clustering. Once enough training data has been collected, the classification can automatically be extended. Importantly, the devices need not be newly introduced into the home, but may simply be previously unmonitored loads. In this way, eNILM addresses a fundamental barrier to practical NILM deployment.

Contributions. We make the following novel contributions:

- The first NILM system design capable of automatically adjusting to the appearance of multiple new devices and device states using density-based clustering, requiring minimal user input in the form of a name tag.
- A new classifier design for event-based NILM based on ensembles of novelty detectors. To the best of our knowledge, these types of classifiers have not previously been evaluated in the context of NILM.
- A comprehensive evaluation of the system design and its isolated components using public datasets, comparing different implementations across a variety of model types.

II. eNILM Design

To tackle the challenge of seamlessly integrating unknown devices into the framework of a deployed NILM system, we have reimagined the *classification* step of the event-based NILM pipeline, as described in Algorithm 1. We now delineate the design of our framework and the rationale underpinning it.

Algorithm 1 Workflow of eNILM

```
Require: event, MoE, known_devices, clustering, outliers, threshold
  event_scores = MoE.classify(event)
  if event_scores.classes() ⊂ known_devices then
      final_class = event_scores.getMaxConfidenceClass()
  else                                  ▷ No Expert claimed the event
      outliers.append(event)
      if outliers.size() ≥ threshold then
          clusters = clustering(outliers)
          outliers.remove(clusters)
          for cluster in clusters do
              MoE.createExpert(cluster)
          end for
      end if
  end if
```

A. Framework

Criteria. To achieve a scalable, extensible NILM system, it should be highly automated, encompassing the detection of new devices, the collection of training data, and decisions to extend the model. The only user interaction that remains unavoidable within the scope of this paper is the labeling of newly detected devices. Finally, the system should facilitate the integration of new devices with minimal remodeling of existing components.

Classifier Design. Previous designs [11], [12] required retraining the NILM models when a new device is added, resulting in considerable overhead. Instead, we employ a mixture of expert (MoE) of density-based outlier detectors, serving as filter for unknown devices and as classifier simultaneously. These experts require only positive class training samples and can decide, based on similarity/density measures, whether a new sample belongs to the positive class, avoiding the need for retraining existing experts when integrating a new class.

Extensibility. Previous research [12] suggests collecting data points that cannot be assigned to existing devices to create a single new class within the model. To address the presence of *multiple* unknown devices and measurement errors (outliers), we opted for a density-based clustering approach instead.

B. Design Choices

As a proof-of-concept for our adaptation, we compared various versions of eNILM. We also implemented the preceding event detection step using publicly available real-world data.

Dataset Selection. For evaluating the proposed framework's deployment applicability, we selected an open-access dataset from a real household rather than a controlled lab environment for a realistic view of the advanced challenges in NILM.

For evaluating the framework and its components, we primarily selected the FIRED dataset [14], which monitors a two-person household over 101 days with 66 devices. The sampling frequency is $2kHz$ for isolated measurements and $8kHz$ for aggregated measurements, which suffices to calculate harmonics-related features and collect ample events.

For further evaluation of the complete framework, we also used the DARCK dataset [15], which includes $1Hz$ readings of the mains meter and all appliances within a two-person household over the span of 6 months. This dataset offers the most realistic view on a real-world deployment, using off-the-shelf hardware for all sensors and a sampling frequency that forbids the use of high-frequency features. Other, more popular NILM datasets either lack in the number of monitored devices or are collected in controlled, artificial environments.

Event Detection. Assuming that relevant changes in a device's consumption during operational state transitions are observable in the aggregated data, our event detection implementation was applied on the mains readings and closely follows established practice, using a probabilistic approach focusing on identifying maxima and minima as potential events [3].

Model Selection. For the extended classification step, a combination of two models is required: a classifier capable of filtering out events of unknown devices and a clustering algorithm that groups events of unknown devices into sets adequate for training the classifier to integrate a new device. We selected Isolation Forest (IF) and Local Outlier Factor (LOF) as classifiers inside the MoE and compared their performance in anomaly detection and classification tasks on FIRED (cf. Sections IV-B and IV-C). Similarly, we employed DBSCAN and HDBSCAN for the clustering task, which effectively manage noise and outliers while producing usable clusters.

Feature Selection. We computed established features on the aggregate data [4] and afterwards performed a forward feature selection using the Gini score of a Random Forest (RF) classifier. The 10 most discriminating feature dimensions

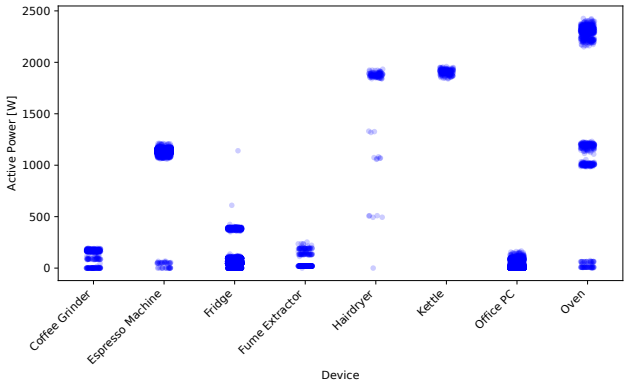

Fig. 1. Power distribution of the devices from FIRED.

according to this procedure were Active Power, Apparent Power, Mean Variance Ratio, Form Factor, the Harmonic Energy Distribution of the line frequency, and the 18^{th}, 19^{th}, 21^{st}, 22^{nd} and 23^{rd} entry of Current Over Time. During our evaluation, we call this feature set the *reduced* feature set.

III. Implementation

We instantiated several variants of eNILM for comparison. After running event detection on the full dataset, we discarded devices with fewer than 10 detected events to ensure a minimum level of behavioral variance and more than one test event per class. For the detailed analysis of eNILM, we selected a subset of 8 FIRED devices whose power distributions overlap at least once (cf. Figure 1), including both low ($<$ 200W) and high power-consuming devices as well as multi-state devices. Because the resulting event distribution is strongly imbalanced, we used SMOTE [16] to oversample minority classes and obtain a balanced training set. Test data was not augmented to avoid bias in the results. To assess the effect of oversampling, we evaluated both the original and augmented training data.

Tuning. We optimized each eNILM component separately on an altered set of devices and features and on a smaller event set to avoid bias from a priori knowledge. For clustering, we maximized the mean accuracy over all possible device combinations, using the mean cluster accuracy under majority-label assignment as the objective. For the MoE classifiers, we evaluated all combinations of up to 4 known devices using random stratified $80/20$ train/test splits without oversampling; all events of the remaining devices were treated as outliers. Hyperparameters were tuned with GridSearchCV and 10-fold cross-validation. For IF-MoE, two near-optimal settings emerged at 100 and 476 $max\_samples$, allowing us to compare accuracy against runtime trade-offs. For HDBSCAN, the two best settings were 10 and 44 for $min\_cluster\_size$: the former was better at producing clusters for all considered devices, while the latter achieved the highest cluster accuracy but missed devices with very few events.

IV. Evaluation

To evaluate the proposed eNILM framework, we first discuss a baseline comparison with related work. Next, we perform an in-depth analysis of the filtering, classification and clustering capabilities. Finally, we assess eNILM, configured with the model choices for generalizability, on two different datasets.

A. Preliminary Investigation

Motivation. To illustrate the issue unknown devices pose to static NILM classifiers, we performed a simple analysis: We trained a baseline RF on half of the 8 devices listed in Figure 1 and evaluated it twice: Once using test data from the devices seen during training, and once using test data from all 8 devices. This process was repeated for all possible device combinations, achieving a median accuracy of 98.5% on the test set containing only known devices. However, when events from unknown devices were included, performance dropped significantly to a median accuracy of 26.4%, highlighting the critical importance of accounting for unknown devices in NILM.

Comparison. A common evaluation scenario in related studies involves detecting 1 unknown device among 4 known devices [6], [7], [12]. Based on this, we evaluated all possible combinations of 4 known and 1 unknown device, using 30 different train/test splits. Our results demonstrate that eNILM, with LOF-MoE, achieved a median F1-score of 94.6%, staying competitive to related work [6], [7], [12]. The IF variants obtained slightly lower F1-scores of 80.8% and 87.6%, while a baseline RF model achieved 88%. However, existing approaches either lack the ability to integrate detected unknown devices or are limited to handling only one new device at a time. eNILM fills this gap by clustering the filtered events and seamlessly integrating them into the existing MoE framework.

B. Filtering

Setup. The first evaluated component is the filter capability of the MoE-classifier. Our preparation revealed two different configurations of the IF-MoE to be comparably suitable in terms of mean novelty detection accuracy, hence, our evaluation includes both these configurations as well as the LOF-MoE.

We evaluated all possible device combinations up to a total of four known devices, chosen from eight devices (cf. Figure 1), with one expert per known device. For training, 80% of the available data points were used, selected with a pseudo-random stratified split, using 100 different seeds. Testing was performed on the remaining 20% of the data for known devices (positive class) and all events of the remaining unknown devices (negative class). An event is only evaluated as negative, if none of the experts classifies it as inlier.

Results. As depicted in Figure 2, the accuracy of the vast majority of test runs remained above 80% for the IF variants, with LOF trailing noticably behind. The TPR showed a different trend, with LOF outperforming one the IF-MoE variants. Further analysis revealed that IF-MoE was particularly skilled at identifying outliers but struggled with inliers, whereas LOF showed the opposite trend.

Compared to related work of detecting unknown devices [3], where the LOF was introduced both standalone and as MoE, our results trail slightly behind: Breyer et al. reported F1-Scores of 93% on a set of four known and four unknown devices,

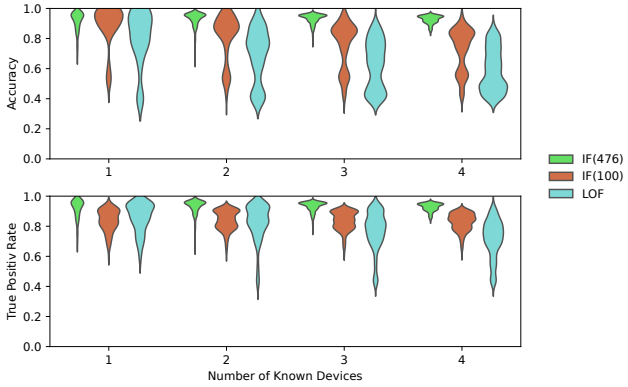

Fig. 2. The accuracy and TPR of different MoE models for identifying events of a different number of known devices out of a set of 8 devices in total.

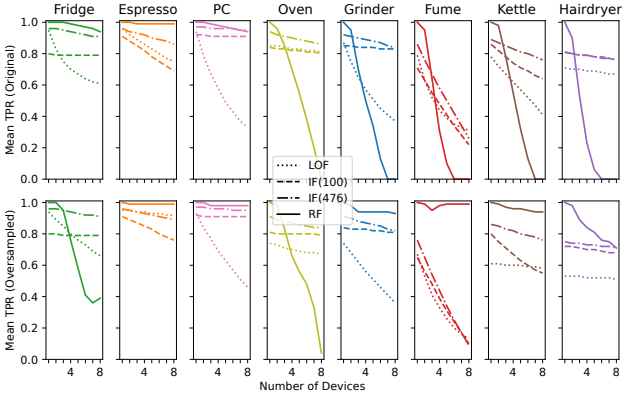

Fig. 3. The TPR of different MoE models for correctly classifying events of known devices out of a different number of devices in total, each compared between a sparse (upper) and an oversampled (lower) training set. The RF classifier serves as baseline.

however, when being tuned with domain knowledge. Whilst some of our experiments were able to outperform these scores, the median scores did not, leaving no generalizable conclusion.

C. Classification

In addition to filtering out events from unknown devices, the MoE-classifier is intended to label events from known devices. This is, to our knowledge, the first proposal of an IF- or LOF-MoE for the multi-class device classification in NILM.

Setup. We conducted tests on all subsets of the 8 considered devices. For each test, the classifier was trained using 80% of the available data, stratified and split with 100 different pseudo-random seeds for reproducibility, with one expert per device. The models were then tested on the remaining 20% of the data. We also compared their performances using oversampled training data with the same seeds. In this case, an event was evaluated as positive, if the expert of the respective device class had the highest confidence score. As a baseline for comparison, we implemented a commonly used RF (e.g., [3]).

Results. As shown in Figure 3, the results vary considerably. The devices are sorted in descending order of the available events, with their mean TPR over all seeds for each model depicted. We observed a similar trend to Section IV-B where the IF-MoE often confuses events from known devices for outliers, sometimes up to 20% of tested events. Conversely, the LOF-MoE also struggles in correctly identifying events for devices with lower event counts. Notably, the RF only performed for devices with a very high number of events and struggled to scale to a higher number of devices to differentiate.

Another important aspect is the slope of the line—the steeper the descent, the more likely a second expert will claim to know the event, resulting in ambiguous labeling. For this aspect, the IF-MoEs are clearly advantageous, as their average performance is in many cases almost independent of the number of devices considered. In contrast, the LOF-MoE struggles significantly with half of the devices, however, not as drastic as the baseline RF that fails to identify many devices as soon as the number of classes exceeds 7. Oversampling generally does not lead to significant improvements for the MoEs but instead for the RF, hinting at the density-based classifiers needing less training data and therefore being suited for faster deployment.

Our investigation of the suitability of density-based MoEs for device classification shows that they tend to scale better to high device counts than state of the art classifiers. Additionally, they require less training data overall, increasing their competitiveness. With the RF struggling on unbalanced datasets, as opposed to the LOF and IF variants, the latter might also require less preparation work whilst staying more suitable to real-world use-cases where class imbalance is to be expected.

D. Clustering

Setup. The clustering aims to create suitable training data for new devices from unclassifiable events. The challenge here is twofold: Firstly, the algorithm does not know the actual number of unknown devices, and hence clusters, during the clustering step. Secondly, the algorithm must account for noisy data and outliers, meaning it is possible that not all events can be assigned to a cluster, or in the worst case, no clusters can be created at all. We evaluated both configurations of HDBSCAN (cf. Section III), as well as the optimal version of DBSCAN, on all possible combinations of up to 8 devices using both original and oversampled data.

Results. As shown in Figure 4, the mean accuracy of all models is relatively high, never dropping below 91%. When using oversampled data instead, the models were on average more capable of scaling to the number of devices. Generally, HDBSCAN creates more clusters than DBSCAN, which is expected due to its design. Instead, DBSCAN fails to incorporate the data of some devices, marking them as outliers instead. On normal data, both HDBSCAN variants have the advantage of discerning low-power-consuming devices, which DBSCAN struggles with due to its global distance threshold.

Our analysis shows that HDBSCAN is generally preferred over DBSCAN due to its flexibility with local densities. DBSCAN struggled to create separate clusters for low-power-consuming devices. Sparse data scenarios favor a low minimum cluster size for HDBSCAN to ensure all devices are detected.

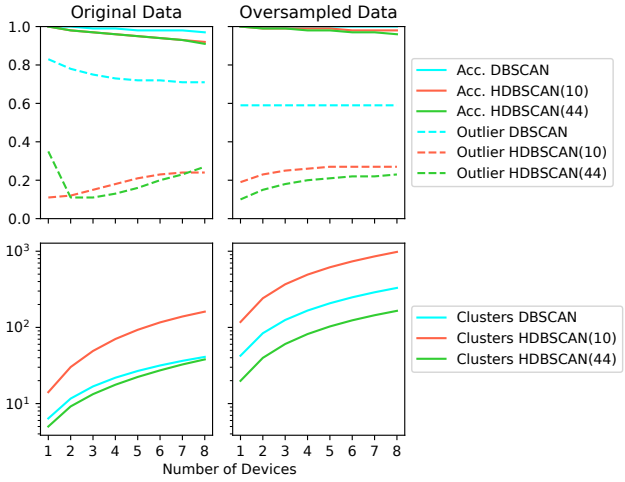

Fig. 4. The mean accuracy, quota of outliers, and number of produced clusters of different clustering models for all combinations of different numbers of devices out of a set of 8 devices in total, each compared between a sparse (left side) and an oversampled (right side) dataset.

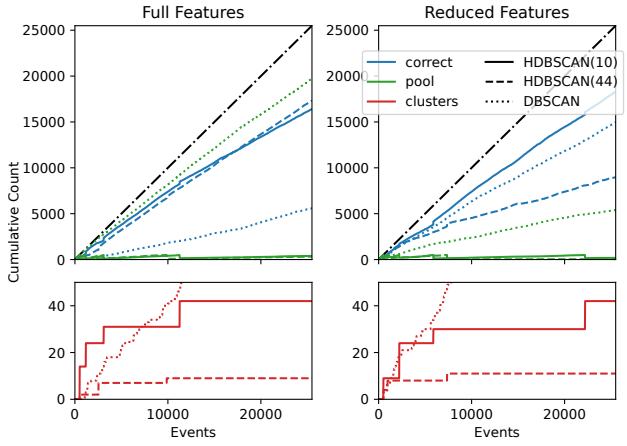

Fig. 5. Cumulative development of the chronological events in FIRED [14] that have been correctly classified or are currently in the training pool for clustering, along with the current number of clusters. A clustering threshold of 500 events is used with LOF using different clustering methods on both the complete as well as the reduced feature set.

In data-rich scenarios, a larger cluster size is advantageous, reducing the number of clusters without loss in performance.

E. Bootstrapping eNILM

Setup. After establishing eNILM's general capability of dealing with multiple unknown loads at once, we set out to examine the most extreme case: Deploying the system in a real environment without any prior knowledge. For an exemplary analysis of eNILM's real-life performance, we chose LOF over the IF configurations, trading better scores for faster runtime. Since the effect of the order of sequentially processed events on clustering and resulting experts is unclear, our preliminary investigation could lead to false conclusions on the best choice of models solely through the resulting scores. For this reason, we also compared all three clustering methods within our eNILM evaluation, using LOF-MoE with each of them.

To evaluate the framework, we sequentially fed all events of all available devices in their actual order as collected in FIRED into the system, using no pre-trained experts. We tested multiple different thresholds with a step size of 50 for the training pool size when clustering was triggered.

Results. Figure 5 depicts the sequential evaluation using a threshold of 500, which performed best. Until the threshold is reached, all events are directly fed into the clustering pool, as no expert exists to assign a label. After about 4 hours of dataset span, the number of correctly classified events suddenly increases for the HDBSCAN variants, as experts are created and their respective events get labeled. We observe a steady increase in correctly classified events, with the pool size staying below the threshold size, indicating that triggered clustering usually results in new experts being created.

As foreshadowed with the clustering analysis (cf. Figure 4), DBSCAN remains mostly incapable of creating meaningful clusters: Even if new clusters are created, usually they only use the minimal amount of data points (12). Subsequent events are seldom claimed by any of the created experts and instead fed into the pool of outliers. This results in the clustering being triggered almost every time any event arrives, leading to a disproportionally high runtime of the framework. The newly created clusters also do not significantly improve the situation.

HDBSCAN with a cluster size of 10 produces more clusters than with a cluster size of 44 (cf. Figure 5), consistent with our observations in Section IV-D. Its number of clusters stays however far below the number of DBSCAN, due to its clusters overall being more effective. Both HDBSCAN variants show comparable classification performances. These observation trends were independent of the chosen threshold.

When using a reduced feature set (cf. Section III), the clustering pool remained small but there was a significant decrease in true positives for HDBSCAN with a cluster size of 44 and a slight increase in true positives for a cluster size of 10. The overall number of clusters remained consistent for the HDBSCAN variants. Although DBSCAN produced even more clusters with the reduced feature set, it was able to increase its performance, surpassing one of the HDBSCAN variants.

F. Transfer Evaluation

Setup. To further investigate the potentials of eNILM, we evaluated it on the DARCK dataset, including 36 devices with a sufficient number of events. However, the majority of these devices exhibited a low quantity of events (≤ 100).

The evaluation of eNILM on DARCK introduced additional complexity not only due to the higher number of device classes but also due to the low sampling frequency and measured quantities: Being restricted to $1Hz$ data of the active power resulted in the feature vector consisting of only active power, max-min-ratio, peak-mean-ratio and form factor [4], as other features would have either required a higher sampling frequency or information on voltage and/or current. We reused the hyperparameters and model types as in FIRED. However, since our comparison of MoE-variants in Section IV-B did not shed

light on the impact of a feature set restricted to low-frequency features, we tried all possible combinations of models. The events were fed into eNILM in order of appearance in DARCK.

Results. The results were a lower TPR than on FIRED, however, still reaching values of above 50% for almost all configurations, with the only exception being the IF configuration. DBSCAN did not suffer from creating a disproportionate amount of clusters as it did on FIRED, with it instead scoring in between the different HDBSCAN variants. Despite the higher amount of devices that could result in new experts, the overall number of experts that were created stayed in a similar range to FIRED.

The macro F1 score of below 15% reflected an issue so far unaddressed: Event-based NILM systems are known to require high-frequency features to scale to a large number of devices. To confirm our suspicion, we trained a k-Nearest Neighbor (kNN) on the same data, which equally is based on notions of distance as the clustering algorithms are. However, for the kNN, we used five-fold cross validation with stratified splits, to ensure representation of all classes in the training as well as the testing data. On average, the kNN achieved a macro F1 score of 22.7%, which is higher but still a subpar score, considering its advantage of knowledge about all classes.

These findings lead us to the conclusion, that generally the transferability of eNILM is given, however, in a low-frequency setting it does not solve the issues of event-based NILM of insufficient feature information. However, this problem is orthogonal and out of the scope of our contribution.

G. Discussion

Our investigations on the proposed eNILM demonstrated its general utility, though with some areas for further investigation. Firstly, the amount of training data for the experts needs to be sufficient, meaning the threshold for triggering clustering should be set carefully. Similarly, a suitable feature set is crucial for eNILM and for event-based NILM in general.

Heading into the more complex scenario of bootstrapping eNILM without prior knowledge, we found the system to be generally capable of learning well-represented appliances. However, it turned out to be incapable of learning underrepresented devices on its own. Future work should explore how pre-trained experts, especially for seldom-used appliances, can enhance eNILM's overall reliability, and if retraining can improve the stability of predictions. Generally, the whole process (cf. Algorithm 1) was completed within less than $1s$ per event on average, making eNILM real-time capable.

V. Conclusion

We introduced eNILM, an extensible framework designed to dynamically integrate multiple new devices into NILM systems with minimal user intervention. Leveraging density-based clustering algorithms, HDBSCAN and DBSCAN, along with novelty detection techniques such as IF and LOF, eNILM adapts to new device signatures effectively. Our comprehensive evaluation using the FIRED and DARCK datasets demonstrated eNILM's capability to handle the dynamic nature of real-world household environments. The system achieved over 85% accuracy in filtering out unknown devices and over 91% accuracy in clustering unknown devices, validating its robustness and scalability. These results highlight the practical applicability of eNILM in real-world scenarios, providing a significant advancement in NILM technology. Overall, eNILM represents an inspiring step in making NILM systems more adaptable and reliable, paving the way for broader adoption.

References

[1] N. W. Arnell, J. A. Lowe, A. J. Challinor, and T. J. Osborn, "Global and Regional Impacts of Climate Change at Different Levels of Global Temperature Increase," *Climatic Change*, vol. 155, pp. 377–391, 2019.

[2] G. W. Hart, "Nonintrusive Appliance Load Monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[3] J. Breyer, M. H. Alizai, S. Samant, and K. Wehrle, "Advanced Filtering of Unknown Devices in Event-Based NILM," *SIGENERGY Energy Informatics Review*, vol. 4, no. 5, p. 8–16, Apr. 2025. [Online]. Available: https://doi.org/10.1145/3727200.3727204

[4] M. Kahl, A. Ul Haq, T. Kriechbaumer, and H.-A. Jacobsen, "A Comprehensive Feature Study for Appliance Recognition on High Frequency Energy Data," in *Proceedings of the Eighth International Conference on Future Energy Systems*, 2017, pp. 121–131.

[5] E. Azizi, M. T. Beheshti, and S. Bolouki, "Appliance-level Anomaly Detection in Nonintrusive Load Monitoring via Power Consumption-based Feature Analysis," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 4, pp. 363–371, 2021.

[6] S. Welikala, C. Dinesh, R. I. Godaliyadda, M. P. B. Ekanayake, and J. Ekanayake, "Robust Non-Intrusive Load Monitoring (NILM) with Unknown Loads," in *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*. IEEE, 2016, pp. 1–6.

[7] J. Zhang, X. Chen, W. W. Ng, C. S. Lai, and L. L. Lai, "New Appliance Detection for Nonintrusive Load Monitoring," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4819–4829, 2019.

[8] L. De Baets, C. Develder, T. Dhaene, and D. Deschrijver, "Detection of Unidentified Appliances in Non-intrusive Load Monitoring Using Siamese Neural Networks," *International Journal of Electrical Power & Energy Systems*, vol. 104, pp. 645–653, 2019.

[9] W. Kong, Z. Y. Dong, J. Ma, D. J. Hill, J. Zhao, and F. Luo, "An Extensible Approach for Non-Intrusive Load Disaggregation with Smart Meter Data," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3362–3372, 2016.

[10] G. Tanoni, E. Principi, L. Mandoliniy, and S. Squartini, "Appliance-Incremental Learning for Non-Intrusive Load Monitoring," in *2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2023, pp. 1–6.

[11] C. Lan, Q. Luo, T. Yu, M. Liang, W. Guo, and Z. Pan, "Analytic Continual Learning-Based Non-Intrusive Load Monitoring Adaptive to Diverse New Appliances," *Applied Sciences*, vol. 15, no. 12, p. 6571, 2025.

[12] J. M. Gillis and W. G. Morsi, "A Novel Flexible and Scalable Nonintrusive Load Monitoring Approach Using Wavelet Design and Machine Learning," in *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2022, pp. 441–445.

[13] M. Kaselimi, E. Protopapadakis, A. Voulodimos, N. Doulamis, and A. Doulamis, "Towards Trustworthy Energy Disaggregation: A Review of Challenges, Methods, and Perspectives for Non-Intrusive Load Monitoring," *Sensors*, vol. 22, no. 15, p. 5872, 2022.

[14] B. Völker, M. Pfeifer, P. M. Scholl, and B. Becker, "FIRED: A Fully-labeled hIgh-fRequency Electricity Disaggregation Dataset," in *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2020, pp. 294–297.

[15] J. Breyer, K. Gützlaff, L. Pompe, and K. Wehrle, "Dataset: Device Activity Report with Complete Knowledge (DARCK) for NILM," in *Proceedings of the 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2025, pp. 407–411.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.