Deep learning with human oversight for time-series data

PhD Thesis

Tamara Sobot

A thesis presented for the degree of
Doctor of Philosophy

Department of Electronic and Electrical Engineering
University of Strathclyde
United Kingdom
03/09/2025

Declaration

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree. The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Abstract

Lately we are witnessing a rapid improvement in the performance of artificial intelligence (AI) for tasks that are time consuming or challenging for humans. However, AI typically relies on large datasets and high-quality labels. Data collection itself is often relatively straightforward, while labelling poses a challenge. Additionally, once deployed, algorithm performance deteriorates if the underlying conditions (i.e. data statistics) change. Moreover, there is a strong initiative for lawful, ethical and robust AI algorithms instead of black-box systems.

To address these challenges, human-in-the-loop approaches such as active learning, interactive learning, and machine teaching have been proposed to optimise handling of training data, i.e., minimise the amount of data that needs labelling without compromising the performance, while including human in the design cycle of algorithms.

In this thesis we first design and test oracle-based active learning frameworks, including various ways of selecting data samples for algorithm training, transferability of algorithms to new environments, and using simplified labels that apply to larger parts of signals instead of per-sample labels. Optimal trade-off between algorithm performance and labelling effort is achieved with the amount of labelled data reduced by 85-95% for the non-intrusive load monitoring problem with regular, fine grained labels; by 82.6-98.5% with simplified labels covering larger signal parts, and 83% for the micro-seismic event detection problem.

Next, we move towards human-in-the-loop active learning approaches, including domain experts in the labelling task during active learning. We address practical considerations of active learning, i.e., existence of an oracle providing absolutely correct labels, variable difficulty of labelling available data samples, and errors introduced during labelling if there is no oracle. We design a stopping mechanism for the active learning process, to avoid

unnecessary labelling. We propose several ways to mitigate introduced errors - using expert's confidence to suppress the effects of labels which are difficult to assign, and using a mechanism to detect potentially wrong labels and send them for re-labelling. We validate the proposed solutions for the non-intrusive load monitoring problem in experiments with three domain experts. The results show that the proposed methodology significantly improves model transferability with labelling effort reduced by 61-93%.

Lastly, we design a machine teaching framework with a hybrid human-machine teacher. The domain expert (human teacher) makes a selection of just several representative data samples to lead the algorithm training process, based on which the machine teacher creates labels and curates the training dataset for learning in stages, resembling real-world teaching. Applied for the problem of micro-seismic event classification, we demonstrate the efficiency of the approach, outperforming the random teacher (F_1 score of 0.64) and active learning (F_1 score of 0.71) approaches with the same labelling effort, achieving F_1 score of 0.78.

The work presented in this thesis aligns with several of the United Nations Sustainable Development Goals (SDGs), promoting peace and prosperity for people and the planet. Research applied to the non-intrusive load monitoring problem aligns with goals 7 - "Ensure access to affordable, reliable, sustainable and modern energy for all" and 12 - "Ensure sustainable consumption and production patterns" by providing users with a clear and easy-to-understand summary of their energy expenses. This will help them see when and how much energy is consumed as well as how its carbon footprint. With this information, users can change their habits and adopt more sustainable practices, ultimately reducing CO2 emissions from their homes. Research on micro-seismic event classification with human oversight will strengthen resilience and adaptive capacity to climate-related hazards and natural disasters such as landslides, supporting SDG 13 - "Take urgent action to combat climate change and its impacts".

Contents

List of Figures

List of Tables

Acronyms

AI artificial intelligence

AL active learning

CNN convolutional neural network

DNN deep neural network

IL interactive learning

kNN k-nearest neighbors

ML machine learning

MT machine teaching

NILM Non-intrusive load monitoring

RF random forest

SDG Sustainable Development Goal

SVM support vector machine

XAI explainable AI

Preface

This thesis presents an exploration of deep learning approaches with human oversight, aiming to reduce the amount of labelled data needed for algorithm training while securing human autonomy over AI system lifecycle. The thesis spans the concepts of active learning and machine teaching.

The first contribution chapter presents the foundational work of this thesis on oracle-based active learning, exploring how to efficiently train deep learning models achieving good performance with as little labelled data as possible. This approach is tested with different types of labels and for different types of time-series signals, including energy disaggregation and micro-seismic event monitoring. Building on this foundation, the second contribution chapter delves into the human-in-the-loop concept within an active learning framework for energy disaggregation. It addresses practical considerations regarding when to stop the learning process and how to manage the imperfections of human-provided input. The third contribution chapter explores the machine teaching concept, with a hybrid human-machine teacher. This innovative approach allows a domain expert to guide the training of a deep learning model for micro-seismic event classification with the help of a machine teacher, to maximally reduce burden on the expert.

I hope that this work contributes to the ongoing discourse in the field and inspires further research on deep learning approaches with human oversight.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisors, Vladimir and Lina, for their invaluable guidance, support and encouragement throughout my research journey.

I also want to thank my friends and colleagues in Glasgow, as well as my family and friends in Serbia, for being there for me.

A special thank you to my husband, Srdjan, for his love and understanding, and to my son, Mihailo, whose laughter and joy have brought so much happiness into my life.

This project has received funding from the European Commission under Horizon2020 MSCA-ITN-2020 Innovative Training Networks programme, Grant Agreement No 955422.

Chapter 1

Introduction

As AI algorithms become more available and widely used, and integrated in many aspects of our lives, ethical concerns are rising. To ensure that AI causes no harm to those using it, European Commission has adopted seven principles for trustworthy AI: (1) human agency and oversight, (2) technical robustness and safety, (3) privacy and data governance, (4) transparency, (5) diversity, non-discrimination and fairness, (6) societal and environmental wellbeing, and (7) accountability [1].

Human agency and oversight are achieved through human-in-the-loop, human-on-the-loop and human-in-command approaches, ensuring that human autonomy is not undermined. Human-in-the-loop-based AI systems assume that human is involved in every decision cycle in the system; human-on-the-loop approaches assume that human is involved in the design cycle and in monitoring system operation; and human-in-command approaches assume that human oversees the overall working of an AI system, including economic, societal, legal, and ethical aspects [1]. These human agency and oversight approaches are usually implemented so that algorithm training works in an iterative fashion, via active learning, interactive learning, and machine teaching [2].

Active learning (AL) is a paradigm that optimises the amount of data that needs to be labelled while not compromising the performance of an AI algorithm [3]. It relies on the assumption that not all data samples bring the same amount of information to model training, so some data samples are more worth labelling and including in the training set than the others. It is implemented in an iterative fashion, choosing the most informative samples in each iteration, and the algorithm improves rapidly in the first several

iterations, ensuring high accuracy with low labelling effort. After the initial jump in performance, the labelling cost usually exceeds the performance gain, so further labelling is not needed. In literature, an oracle is often assumed to provide data labels throughout the process, however, there are several challenges arising from that: a labelled dataset is not available in advance; assuming an oracle usually implies availability at any time, and that all the labels provided are equally easy to get, and also absolutely true. Instead of oracle-based approaches, humans (domain experts or end users) naturally fit into the AL concept – they can be included in the loop to provide labels for informative data samples. However, when domain experts or end users provide labels, they can introduce unintentional mistakes, which hinder the algorithm performance.

Machine teaching (MT) is a paradigm where knowledge is transferred from (human) teacher to an AI algorithm [2]. Similarly to AL, MT works iteratively, but the teacher is in control of the training process. The idea is to mimic how learning works in a classroom - the teacher knows which concepts and in which order to teach to a learner to help them acquire knowledge efficiently. Usually, the teaching starts from clear, concrete, and reliable examples, moving towards more complex ones. Sometimes, teaching a machine can be challenging due to the amount of data that needs scanning and labelling in order to decide what data and in what order should be included in training.

Interactive learning (IL) lies between AL and MT – the control of the learning process belongs to both the algorithm and the teacher, and there is a closer interaction between users and learning systems [2].

AL, IL and MT are especially relevant in applications where raw data can be easily recorded, but labelling poses a challenge. Domain experts are often the only ones who can annotate the data, which can be time consuming and costly, thus hindering the use of data. Also, once trained and deployed, the performance of a deep learning algorithm remains stable for a short period of time in a dynamic environment [4], and then starts deteriorating, usually due to changes in data statistics over time, requiring more labelling and maintenance. Additionally, when transferred to a new environments, performance usually drops due to differences in data statistics.

In this thesis, novel AI with human oversight approaches are proposed. First, an oracle-based AL framework is designed to explore different ways of selecting informative data samples and improving model transferability via AL. The optimal balance between labelling effort and performance is

defined, and a way to determine when to stop selecting data samples is proposed. In addition, the use of weak labels, that are easier to collect as they apply to larger parts of input signals instead of per-sample labels is examined to simplify the labelling process. Then, the focus moves towards replacing an oracle and involving domain experts in the loop. Since experts' labels are imperfect, the effect of errors is assessed, and several ways to mitigate them are proposed - attenuating the effect of labels that are provided with low expert confidence, and detection of potentially wrong labels and returning them for re-labelling. In all these approaches, even though a human is in the loop, the control over the process is with the algorithm - that is, the algorithm chooses samples and leads the training process, and the expert only provides labels. Lastly, a human-on-the-loop MT approach is proposed, giving the expert control over the training without the need for labelling beyond providing several representative data samples, making use of a combination of a human and a machine teacher.

The deep AL approaches presented are developed for applications in climate change mitigation, more precisely, for Non-intrusive load monitoring (NILM) from low-frequency smart-meter measurements, and for micro-seismic event monitoring. In both of these application areas, the data is relatively easily collected. NILM offers detailed insights into electricity consumption, enabling users to adjust their habits to reduce energy usage, and consequently lower greenhouse gas emissions. Data collection is done by conventional smart meters installed in homes, recording aggregate electricity consumption of a household. However, labels, pointing when each appliance is used and how much it consumed, are not so easily obtained. One approach involves installation of submeters for individual appliances, which is expensive and often impractical. Alternatively, an expert with the knowledge of many different consumption patterns of appliances, and the ability to recognise them within a noisy recording, can perform labelling manually, but it is also an expensive and time consuming option.

Micro-seismic event monitoring allows for detection and localization of seismic events potentially causing harm to both people and infrastructure. By providing critical information, it supports management strategies to minimise the risks. Data is gathered by seismometers deployed near the active site. A domain expert then has to manually check all recordings to find and label seismic events, which is an extremely hard task - since the events can be very sparse within a recording, and some of them can be missed due to high levels of noise. Therefore, both areas would benefit from methods to re-

duce the amount of labelling needed to create well-performing AI algorithms. Open access, real-world datasets are used in the studies presented to ensure the reproducibility of the research.

The work in this thesis applied to the problem of NILM resonates with the UN Sustainable Development Goals (SDGs) 7 (Affordable and Clean Energy) and 12 (Responsible Consumption and Production) by providing users with a clear and easy-to-understand summary of their energy expenses. This will help them see when and how much energy they use, as well as whether it's from renewable or non-renewable sources. With this information, users can change their habits and adopt more sustainable practices, ultimately reducing CO2 emissions from their homes. The work applied to micro-seismic event monitoring resonates with SDG 13 (Climate Action), strengthening resilience and adaptive capacity to climate-related hazards and natural disasters such as landslides.

Even though the applications to NILM and micro-seismic event monitoring are considered in this thesis, the proposed approaches can be applied to a wide range of time-series signals where data collection is relatively not expensive, but labelling poses challenges.

1.1 Research motivation and aims

The motivation of this research lies in exploration of how human-in-the-loop approaches can be utilized to improve data efficiency for training and transfer of AI algorithms applied to climate change-related problems, i.e., NILM and micro-seismic event monitoring, while ensuring human autonomy is not undermined during algorithm's life cycle.

Our early work on NILM for dairy farms [5] revealed challenges in the transfer of NILM algorithms across different environments, emerging from different labelling approaches of farmers, for example, one farmer labels one big piece of equipment as one appliance, while another farmer labels it as multiple small subcomponent appliances. In addition, some appliances with the same name have distinct signatures in different locations.

Although transfer learning is a popular approach in NILM, the challenges posed by different labelling practices and distinct appliance signatures across different locations highlight the need to explore additional approaches. In this context, data-efficient human-in-the-loop methods, such as AL, pose a promising solution. AL allows for targeted selection of the small amount

of the most informative data samples which can be labelled by data owners at each location, and then used to update and improve performance of the transferred models.

However, AL comes with its own challenges, such as oracle assumption, unintentional errors introduced during labelling in the absence of an oracle, and availability of the person providing labels. Moreover, in AL the algorithm has full control over the training process, and labelling even a small amount of data in AL can be tedious for the human included in the loop. There is an initiative to give humans more creative and meaningful tasks, and this is where MT can become the preferred choice.

This leads to the following research questions:

RQ1 Can oracle-based deep active learning be useful for efficient training and transfer of AI algorithms applied to time-series data classification?

RQ2 When and how to optimally stop the active learning process? Can an acquisition function with an inbuilt stopping mechanism be designed to be used within an active learning framework?

RQ3 How do errors that humans unintentionally introduce during the AL process affect the performance of the AI model being trained? What can be done to mitigate these errors - can labeller's confidence be utilized? Can incorrect labels be detected and corrected?

RQ4 Can expert's knowledge be used more efficiently through machine teaching with a hybrid human-machine teacher, i.e., can experts be included in the process with a higher level of control over the process, and with less labelling effort?

Chapter 3 addresses RQ1 by designing AL frameworks, exploring if it can improve model transferability; how choice of acquisition function affects the performance, and whether training from scratch of fine-tuning the model yields better results. RQ2 and RQ3 are discussed in Chapter 4: RQ2 is addressed by designing an acquisition function based on hypothesis testing, with a region of uncertainty whose emptiness indicates the end of the AL process; RQ3 by exploring how different amounts and different types of labelling errors affect model performance, by setting sample weight during training based on the expert's confidence when providing labels, and by designing a mechanism to return wrongly labelled samples for re-labelling based

on match between provided label and model output after training. Chapter 5 addresses RQ4 by designing an MT framework for micro-seismic event monitoring with a hybrid human-machine teacher, leveraging on automatic labelling, and allowing domain expert to efficiently lead algorithm training by choosing representative anchor samples based on which training dataset contents are controlled.

1.2 Contribution of Thesis

This research begins with oracle-based AL approaches, followed by human-in-the-loop AL. In both cases, the algorithm is in control over the training process. The thesis ends with an MT approach, giving domain experts more autonomy. In summary, the main contributions are as follows:

- Design and evaluation of the first AL frameworks for the low-frequency model-based NILM and for micro-seismic event detection.
- Definition of the optimal point of an AL process based on performance and labelling effort trade-off, and an acquisition function based on hypothesis testing, with a stopping mechanism to avoid labelling that does not bring significant performance improvement.
- Proposing several ways to deal with errors introduced during labelling by humans included in the AL process: utilizing expert's confidence to weigh data samples for training, and detecting and re-labelling potentially wrongly labelled samples based on match rate between provided label and algorithm output after training.
- Designing a MT framework for micro-seismic event classification with a hybrid human-machine teacher, leveraging both on expert knowledge and automatic labelling, to efficiently transfer knowledge to an algorithm with minimal labelling effort.

1.3 Organisation of Thesis

Chapter 2 provides background on AI with human oversight, active learning and machine teaching, state-of-the-art on NILM and micro-seismic event

monitoring, description of datasets used in this thesis, and definition of evaluation metrics used to measure algorithm performance. Chapter 3 presents research on oracle-based AI approaches applied to NILM and micro-seismic event monitoring, exploring transfers to new environments, evaluating performance of different acquisition functions and training from scratch vs fine-tuning approaches. Chapter 4 presents the next step - human-in-the-loop AL with three NILM experts involved in labelling, exploration of labelling error effects and proposing two ways for their mitigation: weighing training samples based on expert's confidence about provided labels, and detection and correction of wrongly labelled samples based on mismatch between provided label and algorithm output. Chapter 5 presents a study moving towards giving human more control over the process - instead of only providing labels for samples asked by the algorithms, the expert actively leads the training by setting anchors based on which the training set is curated; and the approach leverages on a hybrid human-machine teacher, to efficiently use expert's time to only select anchors, and the tedious work of labelling is performed automatically, by the machine part of the teacher.

1.4 Publications

Journals

1. Todic, T, Stankovic, V & Stankovic, L 2023, 'An active learning framework for the low-frequency Non-Intrusive Load Monitoring problem', Applied Energy, vol. 341, 121078. https://doi.org/10.1016/j.apenergy.2023.121078

2. Sobot, T, Stankovic, V & Stankovic, L 2024, 'Human in the loop active learning for time-series electrical measurement data', Engineering Applications of Artificial Intelligence , vol. 133, no. Part F, 108589. https://doi.org/10.1016/j.engappai.2024.108589

3. Tanoni, G, Sobot, T, Principi, E, Stankovic, V, Stankovic, L & Squartini, S 2024, 'A weakly supervised active learning framework for non-intrusive load monitoring', Integrated Computer-Aided Engineering, vol. 32, no. 1, pp. 37-54. https://doi.org/10.3233/ICA-240738

4. Sobot, T, Murray, D, Stankovic, V, Stankovic, L, 'Hybrid machine

teaching with human oversight for classification of seismograms', Applied Soft Computing. https://doi.org/10.1016/j.asoc.2025.114434.

Conference Proceedings

1. Todic, T, Stankovic, L, Stankovic, V & Shi, J 2022, Quantification of dairy farm energy consumption to support the transition to sustainable farming. in 2022 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE Conference on Smart Computing (SMARTCOMP), IEEE, Piscataway, NY, pp. 368-373, International Conference on Smart Computing 2022, Espoo, Finland, 20/06/22. https://doi.org/10.1109/SMARTCOMP55677.2022.00082

2. Sobot, T, Murray, D, Stankovic, V, Stankovic, L & Shi, P 2024, An active learning framework for microseismic event detection. in 2024 IEEE International Geoscience and Remote Sensing Symposium. IEEE International Symposium on Geoscience and Remote Sensing (IGARSS), IEEE, Piscataway, NJ, pp. 493-497, 2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7/07/24. https://doi.org/10.1109/IGARSS53475.2024.10640569

1.5 Author's Contribution to Publications

Journals

1. Active learning and NILM research, development and testing of the active learning framework for the low frequency NILM problem, paper writing. Supervisory input from Vladimir Stankovic (conceptualization, validation, paper writing, project administration, funding acquisition) and Lina Stankovic (paper writing, project administration, funding acquisition).

2. Active learning, NILM and human-in-the-loop methods research, development, and testing of the human-in-the-loop active learning framework, development of user interface, paper writing. Supervisory input from Vladimir Stankovic (conceptualization, validation, paper writing, project administration, funding acquisition) and Lina Stankovic (paper writing, project administration, funding acquisition).

3. Active learning research, deveopment and testing of the active learning framework for the NILM problem with weak labels, paper writing. First author, Giulia Tanoni, contributed a weak labelling approach for the NILM problem, paper writing. Supervisory input from Emanuele Principi, Vladimir Stankovic, Lina Stankovic and Stefano Squartini (paper writing, administration).

4. Machine teaching and human oversight methods research, development and testing of the human-on-the-loop machine teaching framework with a hybrid human-machine teacher, paper writing. Second author, David Murray, contributed Siamese neural network and related background. Supervisory input from Vladimir Stankovic (conceptualization, validation, paper writing, project administration, funding acquisition) and Lina Stankovic (paper writing, project administration, funding acquisition).

Conference Proceedings

1. Research, development and testing of deep learning models for NILM on dairy farms, data analysis, paper writing. Dataset curation by Jiufeng Shi. Supervosory input from Lina Stankovic and Vladimir Stankovic (paper writing, validation, administration)

2. Research, development and testing of the active learning framework for micro-seismic event classification, paper writing. Dataset curation and background provided by David Murray and Peidong Shi. Supervisory input from Vladimir Stankovic and Lina Stankovic (paper writing, validation, administration)

Chapter 2

Prelimiaries and Background

2.1 Introduction to Artificial Intelligence and Machine Learning

Artificial intelligence (AI) and machine learning (ML) are transformative technologies that have rapidly evolved, redefining how we interact with data and automation. AI encompasses a wide range of techniques designed to mimic human cognition, enabling machines to perform tasks that typically require human intelligence. ML, a branch of AI, focuses on the development of algorithms that allow computers to learn from and make predictions based on data. Deep learning, an advanced subset of ML, utilises neural networks with many layers to model complex patterns within large datasets, significantly advancing fields such as computer vision and natural language processing.

In ML, there are several key paradigms that dictate how models learn from data. Supervised learning requires labeled data, where the model learns to map the input data to known outputs. Common applications include classification (categorising data) and regression (predicting continuous values). Unsupervised learning involves training models on data without labels. The goal is to uncover hidden patterns or intrinsic structures within the data, such as clustering similar data points together. Combining aspects of both supervised and unsupervised learning, semi-supervised learning uses a small amount of labeled data alongside a larger pool of unlabeled data, striking a balance that improves learning efficiency.

To enhance the performance of ML models, several key techniques are em-

ployed. Re-training involves training a model from scratch on a new dataset, allowing it to adapt to changing conditions or requirements. Fine-tuning modifies an already trained model on a new task or dataset with a smaller learning rate, optimising its parameters without starting the training process anew. Incremental learning, also known as online learning, allows a model to continually learn from new incoming data without needing to retrain from the beginning. This is particularly useful in dynamic environments where data evolve over time.

Within ML, active learning (AL) serves as a specialized approach that optimises the training process by selectively querying informative data points, thus improving model performance with fewer labeled examples. Machine teaching (MT) is another approach that enhances ML, allowing domain experts to guide and structure the learning process, ensuring that ML models align more closely with specific tasks and objectives. These approaches are often used to achieve human agency and oversight, as one of the main principles of trustworthy AI.

AL is described in detail in Section 2.3, and MT in Section 2.4.

2.2 AI with human oversight

Human agency and oversight are achieved through human-in-the-loop, human-on-the-loop and human-in-command approaches, ensuring that human autonomy is not undermined during the design and testing of machine learning algorithms. Human-in-the-loop-based AI systems assume that human is involved in every decision cycle in the system; human-on-the-loop approaches assume that human is involved in the design cycle and in monitoring system operation; and human-in-command approaches assume that human oversees the overall working of an AI system, including economic, societal, legal, and ethical aspects [2].

These approaches can be implemented such that algorithm training works in an iterative fashion, via AL, IL, or MT [2]. These approaches differ on who is in control of the learning process: from AL [3], where the algorithm that is being trained, i.e., the learner, is in full control of the learning process requesting labels from the domain expert, i.e., the teacher, based on the learner's confidence; to IL [6], where both the teacher and the learner are in control, and in MT [4, 7], where control lies fully with the teacher, i.e., the teacher selects the most reliable labels to be used for training. Besides

providing human agency and oversight, these methods also enhance data efficiency - being designed to intelligently choose training samples so that the amount of data to be labelled and used for training is optimised.

Conventional machine learning approaches with human oversight, such as [8], include domain experts (humans) in the process of data labelling, explaining decisions of the AI algorithm and refining the AI algorithm through post-processing. However, there is growing interest in going beyond this, and giving humans more meaningful, creative and concise tasks, which would allow for wider human participation in the overall AI system design and maintenance [2]. A promising ways of achieving this are *active learning*, an approach where the AI algorithm identifies the most informative data samples for human labelling, optimising the labelling process and focusing human effort where it is most needed, enhancing the quality of the training data; and *machine teaching*, an approach where human teacher selects, orders and labels data samples, using domain knowledge to optimise the training dataset. By integrating active learning or machine teaching into the workflow, a more dynamic and collaborative environment that leverages both human expertise and machine efficiency can be created. It is important to acknowledge that, even though domain knowledge is often used to denote knowledge of labels in human-in-the-loop approaches, it truly extends far beyond mere labeling, encompassing deep insights into data distributions, feature relevance, task-specific constraints, and underlying generative processes.

2.3 Active learning

The main goal of active learning (AL) [9] is to reduce the amount of labelled data needed to train models. It is an iterative process, where an initial model m_0 is trained using a limited set of labelled data $\boldsymbol{D}_{\text{pt}}$. The prediction is then performed on a large pool of data $\boldsymbol{D}_{\text{pool}}$ where labels are not available, and the acquisition function $q(\cdot)$ is used to select samples $\boldsymbol{Q} \subseteq \boldsymbol{D}_{\text{pool}}$ that are worth including in training, i.e., that satisfy some informativeness criteria, as in [10], diversity criteria [11], or both [12], [13]. Labels are requested for the chosen samples, and after they are available, those samples are included into a new fine-tuning (or re-training) set $\boldsymbol{D}_{\text{ft}}$. When retrained or fine-tuned on $\boldsymbol{D}_{\text{ft}}$, the model uses new knowledge to query more data. The loop runs until a stopping criterion has been met, as shown in Algorithm 1, where algorithm *train* performs either re-training or fine-tuning of the model.

An overview of deep AL, explored recently for various types of problems, such as medical image analysis [14], and natural language processing [15], is provided in a recent survey [3].

Algorithm 1 Active learning

$i = 1$ - active learning iteration
m_i - DNN-based model at iteration i ▷ m_0 - pre-trained DNN model
$q(\cdot)$ - acquisition function
$\boldsymbol{Q}_i$ - set of samples queried at iteration i
$\boldsymbol{D}_{\text{pool}}$ - query pool
$\boldsymbol{D}_{\text{ft}} = \emptyset$ - fine-tuning set
S - stopping criterion met (Boolean flag)
while not S **do**
  $\boldsymbol{Q}_i \leftarrow q(m_{i-1}, \boldsymbol{D}_{\text{pool}})$
  $\boldsymbol{D}_{\text{pool}} \leftarrow \boldsymbol{D}_{\text{pool}} \setminus \boldsymbol{Q}_i$
  $\boldsymbol{D}_{\text{ft}} \leftarrow \boldsymbol{D}_{\text{ft}} \cup \boldsymbol{Q}_i$
  $m_i \leftarrow train(m_0, \boldsymbol{D}_{\text{ft}})$
  $i \leftarrow i + 1$
end while

Acquisition functions

Acquisition function is used to select the most worthy data samples from $\boldsymbol{D}_{\text{pool}}$ to be queried, labelled and added to $\boldsymbol{D}_{\text{ft}}$, by ranking samples belonging to query pool $\boldsymbol{D}_{\text{pool}}$ based on informativeness or diversity criteria [3]. For the classification problem, the model produces a vector containing probabilities that a data sample belongs to each of the possible classes/labels. Common approaches use those class probabilities to estimate model uncertainty (e.g., as in [10]). This approach is commonly referred to as *least confidence uncertainty sampling*, and can be implemented in pool- and stream-based fashion. In the pool-based fashion, all samples from query pool $\boldsymbol{D}_{\text{pool}}$ are evaluated and then the best subset, $\boldsymbol{Q}$, is selected. That is, it is assumed that the whole query pool is available at the moment of query. In the stream-based fashion, data samples are considered to arrive in a stream, and the whole query pool is not available at query time - only the arriving sample can be evaluated, and it can be viewed as operating without a pool since data samples are not saved as they arrive. However, for simplicity, the term query pool is kept

to denote the arriving data stream. Therefore, a predefined informativeness threshold is applied to each data sample as it arrives, and if informativeness of the sample exceeds the threshold, then the sample is considered informative enough and it is included in query $\boldsymbol{Q}$, and otherwise it is not. An example in Figure 2.1 shows how uncertainty sampling works in the pool- and stream-based fashion - the task in the example is binary classification, query pool contains 10 samples out of which 4 are selected for the query.

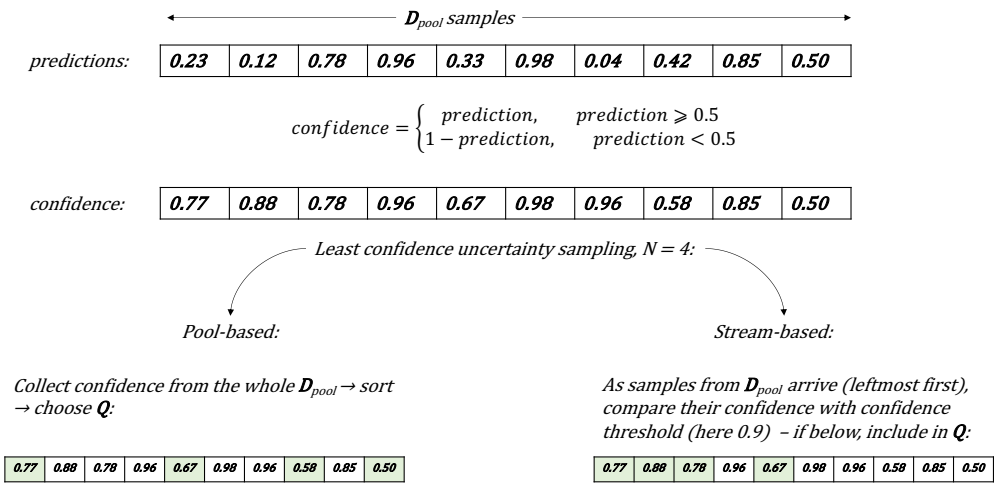


Figure 2.1: Difference between pool- and stream-based uncertainty sampling on an example of binary classification with 10 data samples in the query pool out of which 4 should be selected for query. Samples belonging to $\boldsymbol{Q}$ are highlighted in green.

Selecting a batch of data samples to label independently leads to redundancy because many similar highly-correlated samples would be queried. Therefore, acquisition strategies that account for both informativeness and diversity among queried data samples have been developed. For example, BatchBALD [16] looks at mutual information between a sequence of samples and model parameters. Although it works well with small datasets, it underperforms for large ones [17, 10]. BADGE [11] queries samples that give high-magnitude penultimate layer gradients of different directions if the predicted label would be the true one (i.e., if pseudo-labels are used to compute

gradients). Samples are chosen via k-means++ initialization algorithm on the obtained gradient embeddings. This approach needs computation of gradients for each sample in the query pool, which is resource-intensive. CLUE [12] scales the activations of the penultimate layer of the network with the entropy of the output as uncertainty measure. Obtained embeddings are clustered using k-means algorithm, and then samples closest to cluster centers are chosen. This method depends heavily on the clustering algorithm initialization, and also on the convergence of the clustering algorithm. Acquisition function used in SALAD [13] combines l-2 norms of gradients computed using pseudo-labels as in BADGE [11], and entropy of the prediction as an uncertainty measure. Sum of the two components is greedily maximized to choose samples for query. This approach avoids clustering, but the whole SALAD framework contains pre-trained network, target network, as well as guided attention transfer network, which are all used throughout the process, and which can be demanding.

Stopping criterion

AL is usually performed in an iterative manner, where, in each iteration, the user provides a set of new labels that are used to retrain the model. At some point, newly labelled data supplied to the model will either not anymore improve the performance, or even worse it can start degrading the performance due to overfitting. Hence, it is important to stop the iterative labelling process on time. Setting a threshold on the achieved performance, or observing performance improvement smaller than a threshold [18], can be used to determine when to stop if this is practically possible. Also, confidence levels of the model can be exploited [19], or agreement between the models from a couple of previous iterations [20].

If AL is conducted in small steps (i.e., in each iteration a small number of labelled samples are passed to the model), which is the case in near real-time applications and is the case in this thesis, it is difficult to use stopping criteria based on measuring the improvement between two successive iterations, because small to no improvement can be observed long before the optimal point of AL is achieved. Furthermore, measuring model agreement requires saving either several models from previous iterations, or their outputs for the data used to determine when to stop, which is resource inefficient.

2.4 Machine teaching

Machine teaching (MT) works in an iterative manner, similar to AL, but the training process is led by the teacher, not the model itself as in AL. The teacher is responsible for organising, i.e., selecting, ordering and labelling of the training data set. MT is the preferred technique in applications where labelling is prone to errors, e.g., when it is difficult for a domain expert to confidently assign a correct label, due to noise and interference. In this case, the teacher would label only clean samples instead of, as in AL, providing unreliable labels for the samples that may confuse the learner. The teacher can be a human or a machine [2]. In the former approach, human, i.e., a domain expert, teaches the algorithm high-level knowledge, which is not necessarily an inherent property of the data, but of the human observer, and the learner gets a replica of that knowledge, which is not necessarily complete. In the latter, another machine is the teacher, using an alternative automated method to label data samples, focusing, through an iterative procedure, on finding the minimal number of reliable training examples needed to achieve convergence.

Different implementations of the machine-as-the-teacher framework exist in literature, depending on the level of access the teacher has to the learner's model, i.e., whether the teacher can access the feature space, the loss function and/or the optimisation algorithm of the learner [2]. An Omniscient Teacher can access all the mentioned characteristics, while an Active Teacher has no access to those characteristics, and does not examine the learner directly, but rather queries its performance during the training process and concludes the learner's current status based on the responses, akin to standard human teaching and assessment [2]. A black-box MT paradigm as proposed in [21] assumes the teacher has no access to the learner's model, nor do they share the same feature space, but the teacher makes an estimation of the learner performance instead, and then acts as an Omniscient Teacher. The approach is presented for Least Square Regression, Logistic Regression and Support Vector Machine learner algorithms, however, the optimisation problem used to estimate the learner would be unfeasible for deep learning neural network based learners.

2.5 Areas of application: Non-intrusive load monitoring and Micro-seismic analysis

2.5.1 Low-frequency Non-Intrusive Load Monitoring

NILM [22] consists of breaking down the total power consumption of a building into individual loads. That is, the task of NILM is to estimate the power consumption of individual appliances given only the aggregate power consumption. Formally, the problem of NILM can be described using the following equation:

$$y(t) = \sum_{n=1}^{N} x_n(t) + \epsilon(t) \tag{2.1}$$

where $y(t)$ denotes aggregate power consumption of a building, $x_n(t)$ power consumption of n^{th} appliance, and $\epsilon(t)$ measurement noise. The task of NILM is then to estimate $x_n(t)$ from $y(t)$.

With increased availability of data due to large-scale smart metering roll-out world-wide, low-frequency NILM, where measurements are collected at frequencies below 1Hz, has been dominant in the recent literature, as observed in recent reviews (see [23] for challenges, methods and perspectives for NILM, [24] for a review of deep neural network (DNN) approaches applied to low-frequency NILM, and [25] for NILM solutions for very low-rate smart meter data) due to practicality and low complexity in terms of data management and communication resources.

Low-frequency NILM is a multi-source separation problem [22] in a very low signal-to-noise ratio environment, and hence is particularly challenging in real-case scenarios, due to many similar loads running in parallel in a house, numerous unknown loads, loads changing over time, and measurement noise. Hence, though introduced over 30 years ago, NILM remains a significant research challenge.

NILM methods can be event-based or model-based. In event-based NILM, events, e.g., the moment an appliance is switched on or off, are detected in the aggregate signal in an unsupervised manner (e.g., using adaptive thresholding as in [26] and [27]), and then assigned to known appliances by a supervised classifier (see [28] for a recent review). In contrast, in model-based NILM, a separate model that takes aggregate measurements as input and consumption or on/off state of an appliance as output, is created for extracting power consumption of each appliance, without relying on prior event

detection (see [29] for ana pproach based on factorial hidden Markov models, or [24] for a review of DNN-based approaches.). Although event-based approaches are easier to implement and deploy due to data reduction via extraction of events, they rely heavily on accurate edge detection, and hence are, in practice, susceptible to measurement noise and unknown appliances, causing misclassification of appliances with similar operational power range, as reported in [30] and [31].

From a ML perspective, NILM can be approached as a classification (determining on/off state of individual appliances) or a regression problem (predicting power consumption of individual appliances). In this context, aggregate electricity consumption signals (combined energy usage of all electrical appliances within a household) are used as input to ML models. Input signals are split into windows and fed into machine learning models. Overlapping windows are commonly used (for example, 50% overlap). This improves data utilization, as each timestamp contributes to multiple windows. Also, this helps to capture smoother transitions between different states or load patterns, helping models learn complex behaviors more accurately. Window length depends on the type of the appliance and its cycle length - for example, kettle has a short running time and therefore shorter windows are required (usually a couple of minutes), while washing machine cycle lasts longer, and hence longer signal windows are needed (usually a couple of hours). The way the model processes this input can vary: if a model works in a sequence-to-point fashion, then the output is a single value representing the per-appliance signal for the specific input window; if a model works in a sequence-to-subsequence fashion, then the output is the corresponding sub-window of the per-appliance signal; and if a model works in a sequence-to-sequence fashion, then the output is the whole corresponding window of the per-appliance signal. The output varies depending on the approach - for classification, the output is binary values indicating the on/off state of a single appliance; for regression, the output is the predicted electricity consumption of that appliance.

In classification approaches, NILM classifiers can be implemented as binary or multi-label classifiers. Binary classifiers predict state of only one appliance, while multi-label classifiers predict states of multiple appliances. Binary classifiers are generally more robust - they focus on a single task and operate in a simpler hypothesis space, minimizing potential complications arising from managing multiple labels and their interdependencies. Specifically in the case of NILM, binary classifiers are preferred because of the

nature of data. Each of the appliances present in a household has a unique signature, and the model can focus on that one signature without interference from others. Additionally, when a new appliance is introduced, only one new model for that one appliance needs to be created, without the need to retrain the big multi-classifier model for all appliances. Also, considering transferability, not all houses have the same set of appliances present, making it more practical to have one binary classifier per appliance.

The above described ML-based NILM is illustrated in Figure 2.2 for the example of disaggregation of washing machine electricity consumption, including windowing, processing of the input window, and outputs in case of regression and classification. Overall, the end-to-end processing capabilities and flexibility of machine learning approaches enable effective analysis and prediction of individual appliance behavior based on aggregate electricity consumption data.

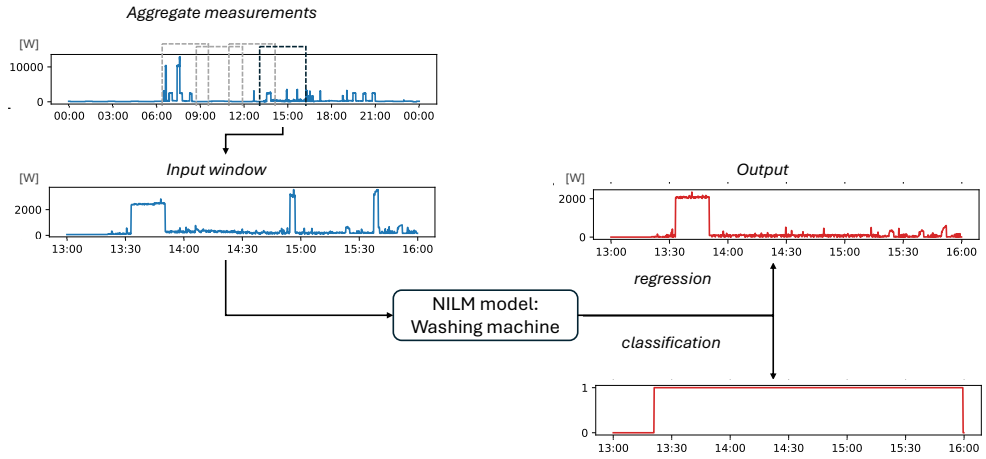


Figure 2.2: Illustration of working of a model-based NILM.

2.5.2 Micro-seismic signal analysis

The usual goal of microseismic monitoring is to detect, locate and characterise microseismic events to provide geometric and more general information about

the considered subsurface processes [32]. Continuous monitoring produces extensive seismic records that may include multiple sources, which require classification. Since many applications require real-time processing, machine learning plays an important role in micro-seismic event classification (see a recent review [32] for a summary of machine learning approaches in micro-seismic monitoring and analysis).

The task of micro-seismic event detection is to find when an occurred, and the task micro-seismic event classification is to assign a categorical label, i.e., event class to a detected event. From a machine learning perspective, the problem of micro-seismic event detection and classification is similar to the problem of NILM - it can also be described as a blind source separation problem. Long recordings from seismic arrays are split into windows and fed to machine learning models as input. Window length depends on the characteristics of the signals that need be captured - for micro-seismic events several seconds is usually enough [33], while, for example, for earthquakes a window length of a minute is frequently used [34]. In case of detection, the output is a binary signal indicating where an event occurred. In case of classification, the output is a class to which the input signal belongs (e.g., quake, rockfall, noise). The task of event detection can be described as:

$$y = \begin{cases} 1 & f(w) \geq \theta \\ 0 & f(w) < \theta \end{cases} \tag{2.2}$$

where y is the output, w is the input signal window, $f(\cdot)$ is a discriminative statistic (i.e., STA/LTA ratio, or neural network score), and θ is a detection threshold. The task of event classification can be described as:

$$y = \arg\max_{c \in C} p(c \mid w; f) \tag{2.3}$$

where y is the output class label, and $p(c \mid w; f)$ is the posterior probability of class c given the input signal window w, parametrized by f (e.g., neural network weights).

The above described ML-based micro-seismic detection and classification are illustrated in Figure 2.3, including windowing, processing of the input window by a ML model, and outputs in case of detection and classification. Although similar to the NILM problem, this is a more complicated case due to very high sampling frequencies. Higher sampling frequency captures wider bandwidth, integrating more total noise power. This causes very low signal-to-noise ratios, making it easy to overfit machine learning models to noise

instead of useful signal patterns. Also, labels are usually created by searching through recordings and manually assigning classes to detected events, producing very unreliable and sparse labels.

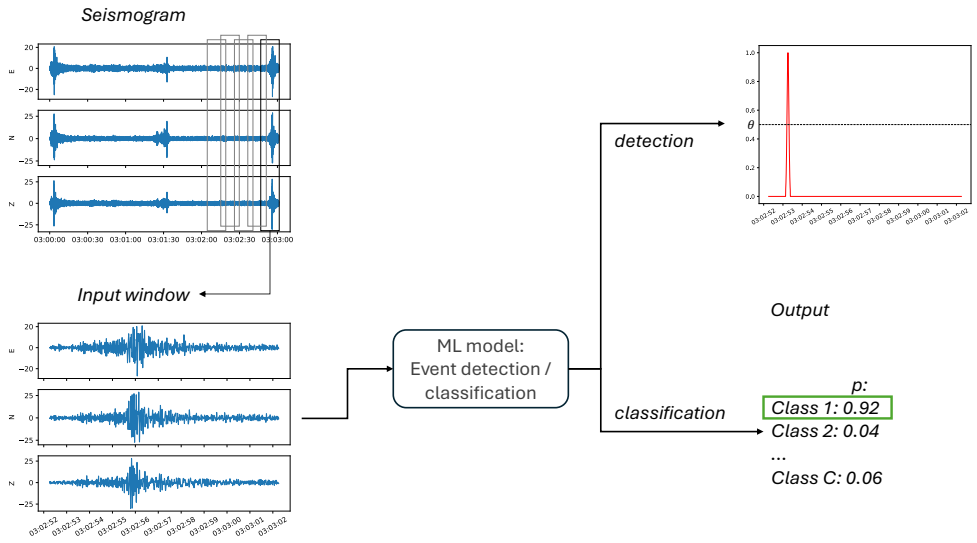


Figure 2.3: Illustration of working of ML-based micro-seismic event detection and classification.

2.6 State-of-the-art: Active learning for time series data

This section provides a general state-of-the-art in AL for time series data. Then, background and state-of-the-art in deep learning approaches, transferability, and AL for NILM is given, followed by background and state-of-the-art in deep learning and AL for seismic signal analysis.

2.6.1 Active learning for time series data

AL has only recently been introduced for time series data to solve anomaly detection tasks [35] and [36]. Two publicly available time series datasets used in [35] have significantly shorter lengths of recordings (5 weeks and 7 days,

respectively) than those used in our study (approximately 2 years). Acquisition functions used in [35] include uncertainty sampling, interval random sampling, and top-k sampling based on abnormality score, and a combination (union) of them all. Stream-based sampling was not investigated.

Anomaly detection with AL and two contrast variational autoencoder (VAE)-based models is proposed in [36]. Combination of anomaly scores and standard deviation of posterior distribution at each point for both models is used for choosing samples in the AL process. However, VAEs do not capture time dependencies in data, which makes them not optimal for time series data. Moreover, this method trains three models in total - two VAEs and one query model which is trained to choose samples based on autoencoders' outputs, which makes the method complex, and hence, training phase is done offline. Our methods use a single DNN model, capable of capturing temporal patterns, that performs both time-series classification and selection of samples to be queried, which makes it more convenient for full online deployment - for both inference and fine-tuning phases.

An integration of transfer and active learning for time series prediction is presented in [37]. The settings are different to ours - in [37], AL is used to choose samples from source domains, which are most suitable for transfer to a target domain with *known data distribution*, while we consider a more realistic scenario, when source domain data are labelled and available, and we adapt the model to a new environment with *unknown data distribution*, and labels have to be queried since they are not available in advance. Time series classification with AL for applications in learning of driving trajectories is presented in [38]. A support vector machine (SVM) and a fully connected neural network were used, for classification of a data point in a latent space for each trajectory. Classes used are balanced, while in this thesis, highly imbalanced datasets are used. Although data used in [38] is time-series, stream-based acquisition functions suitable for online learning were not considered. In this thesis, in Chapter 3 and 4, we explore stream-based uncertainty sampling, which, besides being convenient for time series, has the advantage of online implementation since it does not require the whole query pool to be available in advance.

2.6.2 Non-intrusive load monitoring

DNN-based low-frequency NILM and model transferability

Numerous machine learning approaches have been used in the past (see Introduction and survey papers such as [39] and [40]), with DNN-based methods dominating current literature, due to their very good performance (see e.g, [41] and [42] for comparisons between traditional and DNN-based NILM approaches), flexibility and ease of use (once the models are trained).

A recent review paper [24] summarizes DNN-based low-frequency NILM approaches, concluding that the use of convolutional layers in neural networks has gained in popularity recently - [43] proposes a fully convolutional DNN for a fast sequence-to-point implementation; [44] proposes a convolutional neural network (CNN) architecture, designed to be a generalized network which performs well when transferred to a new domain; [45] proposes a sequence-to-subsequence learning using a CNN; [46] proposes a scale- and context-aware neural network containing convolutional layers; and [47] proposes a CNN and multilabel classification. Recurrent neural network elements in [44] and [48], and newer concepts, such as generative adversarial networks (GANs) in [45] and [46] and attention mechanisms in [49] and [50] have also been attempted. The best performing approaches are the ones using convolutional layers, adversarial losses, multi-task learning and post processing techniques.

Transferability of DNN-based NILM models, i.e., their adaptability to new conditions using user feedback and continuous learning approaches, as well as privacy preserving issues are identified to be key challenges of the current NILM state of the art [23]. The ability to use existing models, or adapt them efficiently to new, unseen environments with dynamic environmental factors and end-user patterns of use, is very important to enable large-scale NILM applications.

Transferability of two DNN architectures across three publicly available datasets - REDD [51], REFIT [52] and UKDALE [53] is explored in [44], without adaptation to new environments. Transferability was successful, though a drop in performance was observed compared to when training and testing with the data from the same dataset. Transferability with adaptation to a new environment is explored in [54], as well as in [55] and [50] - DNN models are fine-tuned using labelled data from new datasets. In [56], cross-domain and cross appliance transferability is investigated, concluding that if statistics of power consumption are similar between different domains, fine-tuning

is not required. Transferability of NILM model in industrial settings, on dairy farms, is tested in [5], concluding that different labelling practices and different appliance signatures in different farms hinder model transferability.

Although transfer learning for NILM has drawn attention recently, many challenges still remain. When transferring a pre-trained DNN-based NILM model to a new environment, the performance is likely to drop significantly. On the other hand, availability of good quality and large amount of labelled data from new domains is assumed when using fine-tuning approaches. In practice, obtaining such labelled data from a new environment requires submetering or manual annotation via a time diary, both of which are resource intensive.

Active learning for NILM

Recent reviews of NILM, including state-of-the-art NILM data sets, feature engineering, as well as learning approaches for NILM, are presented in [28, 24]. Although new and relevant techniques such as transfer learning and federated learning are discussed, AL has not been mentioned. This is mainly due to the very limited amount of work on the topic of AL for NILM with only few initial studies published so far, all focusing on the methodologically very different, high-frequency NILM problem. One of the first attempts to apply AL to high-frequency NILM [57] uses a k-nearest neighbors (kNN) classifier trained on BLUED dataset [58] to identify which activation belongs to which appliance. An AL framework where the algorithm intelligently selects instances for queries based on an informativeness measure, Euclidean distance of the samples in the feature space, is compared to the scenario where the algorithm randomly selects instances to query. The impact of different probability- and distance-based query strategies as well as the choice of the initial training set for event-based high-frequency NILM is investigated in [59]. The performance is evaluated using cross-dataset validation with BLUED dataset [58]. A combination of semi-supervised learning and AL is proposed for training a random forest (RF) classifier for event-based high-frequency NILM on high frequency BLUED dataset [58] in [60]. The results show that including AL outperforms the used semi-supervised learning approach. An active deep learning approach is used in [61], also for an event-based NILM, where a combination of three high-frequency NILM datasets, PLAID [62], WHITED [63] and COOLL [64] with discrete wavelet transform are used to extract high-dimensional appliance features from original

current signals.

From the above, one can notice that the AL approaches for high-frequency NILM (sampling rate in order of kHz) yield promising results, but under some impractical constraints, e.g., AL frameworks for event-based NILM using high-frequency measurements are proposed with the assumption that perfect event detection exists.

The work presented in this thesis in Chapter 3 is the first approach of AL for model-based low-frequency NILM, which are more popular now due to their good performance and practicality due to smart metering roll-out, as per [24] and [28]. The approach presented in Chapter 3 (and in the papers reviewed above) is designed for residential NILM. A recent paper [65] follows this research and proposes an AL approach for NILM in industrial settings, using HIPE dataset. This new contribution broadens the applicability of AL in NILM, and highlights the growing importance of efficient energy management in industrial environments.

2.6.3 Micro-seismic signal analysis: Deep learning-based human-in-the-loop approaches

Deep neural networks tend to be dominating recent literature for micro-seismic analysis due to their ability to perform tasks in an integrated, end-to-end manner [32], as opposed to pipeline-based traditional algorithms (e.g., hidden Markov models, support vector machines, random forests - see [66]). For example, of relevance to this thesis, three CNN-based multi-label classifier architectures, based on time-domain, short-time Fourier transform and continuous wavelet transform, are described in [33], for classification of earthquake, rockfall and low signal-to-noise ratio quake events. To leverage on both temporal and spectral features, an auto-encoder-based deep neural network with attention mechanism, fusing time- and spectral-domain features for rockfall and earthquake detection is proposed in [67]. These approaches try to embed the temporal and spectral features geoscientists consider jointly when labelling, but without involving geoscientists in the AI system design. While such approaches report high classification performance, all operate as black-boxes and all require large labelled datasets for training.

There are only a few early attempts of machine learning approaches with human oversight for seismic signal analysis, mainly for seismic image interpretation. Approaches for time-series signal analysis are rather limited,

based on expert-labelled datasets. A human-on-the-loop approach for seismic recording labelling, verification and re-labelling via a multi-class CNN supported by explainable AI tools is presented in [8]. Training samples are manually chosen by a domain expert, resembling MT with a human teacher; however, the main role of the domain expert is in refining the event catalogue after the model is trained using all the training data at once. It does not avoid the issue for the requirement of a large labelled dataset. An AL approach for seismic stratigraphic interpretation [68] (specifically seismic image semantic segmentation) uses a combination of deep clustering and uncertainty sampling to choose data samples which are then annotated by geological expert, and included in training of an autoencoder deep neural network. Although an improvement in performance is demonstrated with limited data, label queries are made to an oracle (i.e., a simulated expert), whose task is to provide annotations, and who has no control over which data samples are included in the training set, so, domain knowledge is not fully and efficiently utilized. An AL framework for micro-seismic event detection from time-series seismometer measurements is presented in Chapter 3, aiming to reduce labelling effort for training of deep learning-based seismic event detection algorithms. The method is developed for time-series seismic signal analysis; even though it is shown that labelling effort is significantly reduced via AL (up to 83%), an oracle is assumed again to provide labels for data samples queried by the model. So, in a real-case scenario, a domain expert would annotate data samples without any control over the contents of the training dataset. Another AL method, for volcano-seismic event classification from time-series measurements via a CNN, is proposed in [69]. Performance gain when using AL compared to randomly selecting training data samples is achieved for one of the two datasets used in experiments, concluding that AL brings more benefits for datasets with less separable classes, and, as in other reviewed approaches, an oracle provides sample labels without control over contents of the training data set, and the order of training data samples.

In summary, only AL approaches for seismic signal analysis have been proposed so far, all with the same limitations - the domain expert is only asked to label the samples during the learning process, but has no other means of control over the training. This way, the domain knowledge is not used efficiently - there is lack of control over the algorithm training, and the expert still needs time to provide all the needed labels. To the best of our knowledge, no attempts of MT with a human-on-the-loop oversight approach for seismic signal analysis have been attempted.

2.7 Datasets

2.7.1 Non-intrusive load monitoring

Low-frequency smart meter datasets provide insight into energy consumption patterns, offering great potential to reduce electricity usage. The signals in these datasets are characterized by sampling frequencies of 1 Hz or less. This low temporal resolution means that rapid changes in electricity consumption cannot be captured, causing confusion between appliances (e.g. dishwasher activation mistaken for washing machine) and poor transferability. Assuming that confused appliances are predicted with higher uncertainty, these cases pose a valuable source of model improvement though AL.

REFIT and UK-DALE: Smart meter datasets

To facilitate reproducibility of our research, we use the well documented public REFIT [52] and UK-DALE [53] real-world electrical load measurements datasets as these two datasets are among the most widely used datasets for evaluation of NILM algorithms mimicking well real-world conditions [28, 24, 23]. For example, both REFIT and UK-DALE datasets are used in [70] for complexity reduction and transferability via transformer-based architecture, in [56] for cross-domain and cross-appliance transfer, and in [44] for evaluation of transferability of DNN architectures. These datasets are used in Section 3.2, and in Chapter 4. REFIT is also used in Section 3.1.

REFIT consists of 2-year long (2013-2015) continuous time series electricity consumption recordings from 20 houses in the United Kingdom. Each house data contains aggregate electricity consumption time series measurements (see Fig. 2.4), as well as consumption of 9 individual appliances, measured at an 8-sec interval. The large number and diversity of appliance waveforms or signatures across 20 houses makes the REFIT dataset one of the most challenging NILM datasets and a good exemplar for robust evaluation of AL methodologies. An example of recordings from REFIT House 2 is shown in Figure 2.4.

UK-DALE contains recordings from 5 houses in the United Kingdom. Aggregate power is sampled at 16kHz, while appliance power is sampled at an 6-sec interval. Four houses were monitred for a year and a half, and the fifth house was monitored for 655 days.

To align with the widespread smart meter roll-out with in-house recording

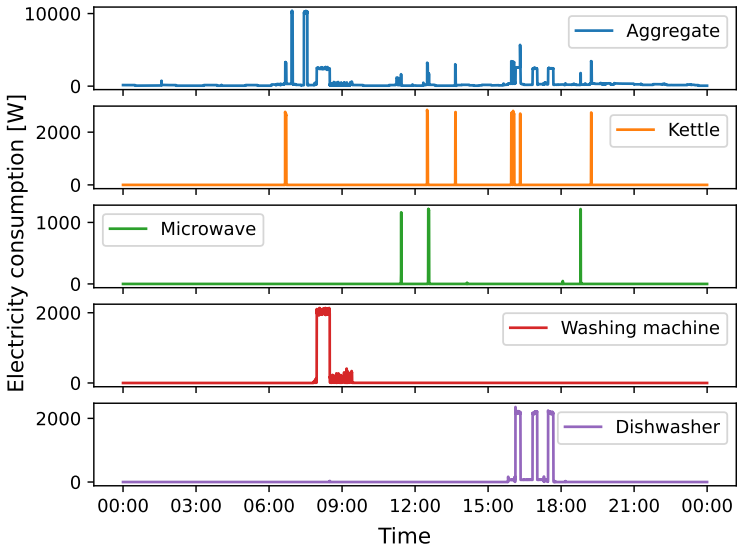


Figure 2.4: Example of electricity consumption recordings from REFIT House 2.

granularity of about 10 sec [71], the data is re-sampled to 10-sec sampling interval for experiments in this thesis. Appliance types used in this study are kettle, microwave and toaster - resistive loads with short activation times - as well as washing machine and dishwasher - inductive (and also resistive) loads, with long cycle duration and multiple states. Examples of typical load profiles for these appliances are shown in Figure 2.5.

Measured aggregate electricity consumption expressed in Watts (W) is normalized using Z-normalization technique: $Z = \frac{x-\mu}{\sigma}$, where x denotes the original measurement, and μ and σ stand for mean value and standard deviation of x across the training dataset, respectively. To determine the ON-OFF state of appliances, thresholds are applied to measured electricity consumption of each appliance, according to Table 2.1. That is, if the appliance consumption value is above this on-power threshold, then the appliance is considered to be turned on, and otherwise it is considered to be turned off.

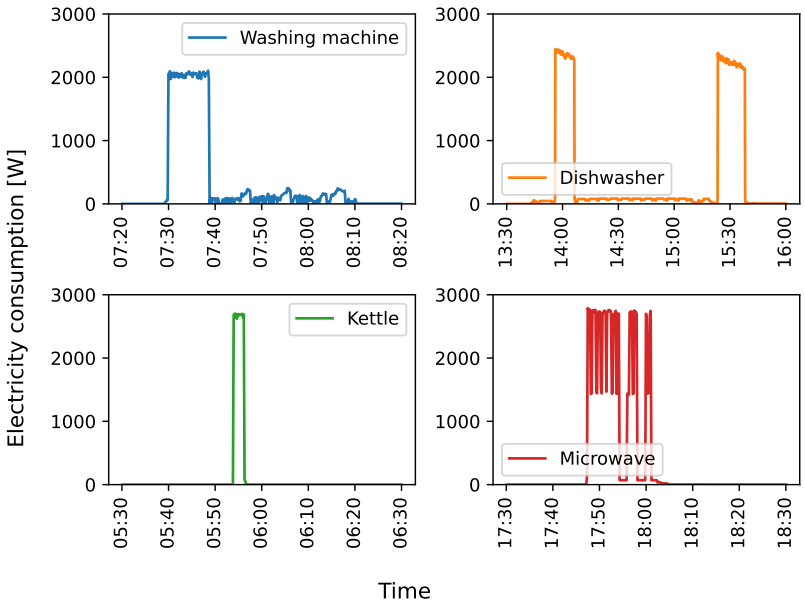


Figure 2.5: Example load profiles for washing machine, dishwasher, kettle and microwave from REFIT House 5.

Appliance	"On-power" threshold [W]
Kettle	2000
Microwave	200
Toaster	50
Washing machine	20
Dishwasher	10

Table 2.1: On-state power thresholds [W] for each target appliance.

2.7.2 Micro-seismic signal datasets

Compared to the smart meter datasets described above, for the NILM problem, the following datasets of micro-seismic signals are more challenging due to high sampling rates used - high-frequency data is more susceptible to noise, which complicates the learning process - deep learning algorithms may overfit to noise instead of learning meaningful signals. Due to the fact that these signals are generally noisy and that the vast amount of data is generated from seismic recordings, labelling is often incomplete and inaccurate, as it

often has to be done manually.

A study evaluating model transferability between Résif and a geographically distinct seismically active site in Larissa (Greece) [33] demonstrates good transfer for catalogued earthquakes, however, it revealed missed earthquakes and false positive quakes and rockfalls among additional, manually checked events. Assuming that the model shows uncertainty for missed or false positive events, they pose a valuable source for improving the model through AL, making it more robust to noise and variations in the data.

Utah-FORGE: Human-induced seismic signals

The dataset used in Chapter 3.3 is recorded as part of the Utah Frontier Observatory for Research in Geothermal Energy (FORGE) project, designed for research on creating, sustaining, and monitoring enhanced geothermal systems (EGS) [72]. Hydraulic stimulation was conducted in the target heat reservoir to create fractures and increase the rock permeability in the hot dry rock. Pressurised fluids are injected into the deviated stimulation well 16A-32 at a depth of around 2.4 km at three different stimulation stages. The stimulation at Utah FORGE is monitored via seismic instrumentation deployed in deep boreholes. During the last stimulation, three deep vertical monitoring wells (58-32, 56-32, and 78B-32) distributed in different azimuths were in place to capture the stimulated fractures, i.e., induced microseismic events. Well 58-32 and well 78B-32 contain 8-level 3-component digital geophones (represented as G1 to G8), with each level separated by 100 feet. In well 56-32, however, measurements are acquired using a two-level analog seismic monitoring tool (G1 and G2). In this paper, we analyse one hour of continuous data recorded at a sampling rate of 4 kHz (from 20:00 to 21:00 on 21 April 2022) recorded by the 18 sensors from the three monitoring wells during the third stimulation stage.

Monitored events are of induced microseismic nature, they are pulse-like and have high-frequency components. To remove low-frequency noise from machinery at the site, and also high-frequency measurement noise, obtained signals are filtered using a fourth-order Butterworth band-pass filter, with the passband between from 100 to 1800 Hz. An example of an event from well 58-32, the bottom geophone (G8) is shown in Figure 2.6. Data is filtered as described above. FP1 and FP2 are horizontal, and FPZ is the vertical seismogram component. Horizontal axis represents time (in samples; 1 sample = 0.25msec), and vertical sensor measurements in mm/sec. Ground truth indi-

cates the time between P and S wave arrivals, marked red in Figure 2.6. An event catalog [73] is generated with the EQ-Transformer of [74], pretrained using a global distribution of earthquakes. It contains timestamps of P and S wave arrivals associated with each event for each sensor where it was picked, together with corresponding signal-to-noise (SNR) values. After automatic label generations with Earthquake Transformer, the predicted labels were validated by seismologists via visual inspection.

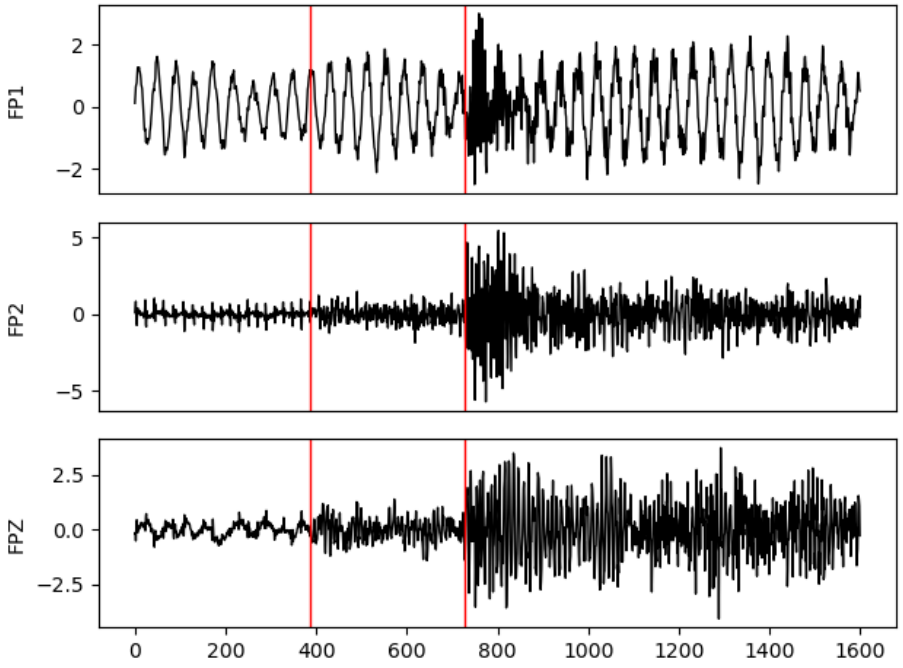


Figure 2.6: A micro-seismic event example. FP1 and FP2 are horizontal, and FPZ is the vertical seismogram component. P and S wave arrivals are marked in red.

Résif: Naturally occurring landslides

An open access dataset from the Résif Seismological Data Portal, recorded by the French Landslide Observatory Observatoire Multidisciplinaire des Instabilités de Versants (OMIV) [75] is used in Section 5. The data is acquired by Super-Sauze C station (from MT network), located east and west of the

Super-Sauze landslide in Southeast France (lat. 44.34787, long. 6.67805). The signals are recorded during 3 periods: 11 October - 19 November 2013; 10 - 30 November 2014; and 9 Jun - 15 August 2015. Data from 3-component sensors are used, originally at a sampling frequency of 250Hz. To filter out noise from human activities, animals and rain, seismograms are filtered using 4^{th} order Butterworth band-pass filter ranging from 3 to 40Hz. Data is normalized using z-score technique (subtracting mean and dividing by standard deviation).

The dataset is accompanied by an event catalogue created by [76]. There are 4 event classes present in the dataset - low-magnitude earthquake (denoted as S) - 335 instances, quake (Q) - 207 instances, rockfall (R) - 351 instances and noise (N) (of anthropogenic/natural origin) - 302 instances.

2.8 Evaluation metrics

Deep learning algorithms used in experiments in this thesis are designed for classification tasks, so their performance is evaluated using the standard F_1 score, which is calculated as the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2} \cdot (FP + FN)} \tag{2.4}$$

where TP denotes true positives - both model prediction and ground truth are positive; FP for false positives - prediction is positive but ground truth is negative; and FN for false negatives - prediction is negative while ground truth is positive.

If multiple classes are present in the dataset, weighted F_1 score can be used as an overall performance measure across all classes present, accounting for class imbalance, as per Equation 2.5:

$$F_{1,weighted} = \sum_{c \in \boldsymbol{C}} \frac{\mid \boldsymbol{D}^c_{test} \mid}{\mid \boldsymbol{D}_{test} \mid} \cdot F_{1,c} \tag{2.5}$$

where $\boldsymbol{D}_{test}$ denotes samples belonging to the test set, $\boldsymbol{D}^c_{test}$ denotes test samples belonging to class c, $F_{1,c}$ denotes F_1 score for class c, and $\boldsymbol{C}$ denotes all classes present in the dataset.

Also, micro-F_1 score, considering the total of true positives, false negatives and false positives (doesn't matter to which class each of them belongs) can

be used if multiple classes are present, as per Equation 2.6:

$$F_{1,micro} = \frac{\sum_{c \in \boldsymbol{C}} TP_c}{\sum_{c \in \boldsymbol{C}} TP_c + \frac{1}{2} \cdot (\sum_{c \in \boldsymbol{C}} FP_c + \sum_{c \in \boldsymbol{C}} FN_c)} \tag{2.6}$$

where TP_c denotes true positives for class c, FP_c false positives for class c, and FN_c false negatives for class c.

Chapter 3

Oracle-based active learning for time-series classification

This chapter presents research on AL with an oracle assumption, that is, ground-truth labels are used throughout the AL process. The significance of this approach is in the potential to improve model performance with very small amounts of labelled data and enhance transferability across new domains.

First, in Section 3.1, an oracle-based AL framework is developed for time series classification, demonstrating the effectiveness of the approach for the NILM problem, with low sampling frequency (0.1 Hz). In this section, the labels provided by the oracle are of the same granularity as the input signal (i.e., each timestamp of the input signal is labelled). The framework is tested with various acquisition functions. In addition, the transferability of models through AL is demonstrated, and optimal point of the AL process is defined. Building on this foundation, in Section 3.2, the framework is applied to the NILM problem with weak labels, i.e. labels that apply to a larger part of the signal (one label covers multiple timestamps). Weak labels are easier to obtain, as they do not require domain knowledge, and can be set by end-users in practice. Section 3.3, extends the framework to the problem of micro-seismic event detection, demonstrating effectiveness of the approach for much more challenging high-frequency data (sampling frequency 4 kHz) with low signal-to-noise ratio, which brings risks of models overfitting to noise instead of learning useful signal patterns.

The framework is shown to be effective, robust, improving the transferability of models to new, unseen domains, and available to use with

weak labels.

3.1 An active learning framework for the low-frequency non-intrusive load monitoring problem

In this section we propose the first active-learning based method for low-frequency model-based NILM, that can operate at scale using smart meter measurements. As opposed to the research already conducted on AL for NILM reviewed in Chapter 2, our approach uses low-frequency measurements and model-based NILM method, with a separate model trained for each appliance disaggregated, eliminating the need to introduce impractical assumptions of perfect event detection. In particular, we leverage on the Wave-net NILM approach of [43], as one of the currently best performing models reported in the recent comparative study [24]. We note that though [43] is used to showcase the proposed methodology, other DNN-based NILM solutions, such as deep neural networks from [77], sequence-to-point convolutional neural networks from [78], recurrent neural network from [79], convolutional and gated recurrent unit-based neural networks from [44], a hybrid of a convolutional and a recurrent neural network from [48], or one-to-many CNN architecture from [80], can be used instead with the proposed AL methodology.

We explore different approaches of selecting the most critical samples to label, i.e., acquisition functions, and discuss their limitations and effect on accuracy and transferability. In the aforementioned AL approaches - [57] using a kNN classifier, [59] with an SVM classifier, [60] using an RF classifier, and [61] using a DNN, high-frequency, event-based NILM methods are used with classic uncertainty-based acquisition functions, which yield one data sample at a time. Since DNN methods process a batch of data samples at a time, it is necessary to group the samples before labelling. Creating a batch of samples by simply joining individually queried samples will likely result in samples that are very correlated; this reduces the effectiveness of learning, since for the model to learn more effectively, it is important that it learns from diverse data. For that reason, we explore BatchBALD [16] which can choose a diverse batch of samples but can be computationally demanding [81].

We consider three practical scenarios in terms of availability of labelled

data, and analyse how the proposed methods perform in various scenarios. We perform a sensitivity analysis w.r.t pre-set hyper-parameters. We discuss optimal performance-complexity trade-off and determine whether complexity can be reduced without performance loss by not re-training the entire model after each interaction, as is commonly done in existing approaches - re-training a k-NN classifier in [57], an SVM classifier in [59], an RF classifier in [60], or a DNN in [61].

3.1.1 Methodology

In this section, the proposed workflow of the AL framework for the model-based low-frequency NILM is described. Given a dataset of aggregate electricity consumption measurements, the goal is to train a DNN-based model (active learner) to predict the on/off state of an appliance at each timestamp while using a minimal number of informative labelled data samples. The framework comprises: (i) the formulation of the training set, query pool and testing set, (ii) the various acquisition functions being explored and (iii) the DNN-based NILM model used to showcase the proposed AL methodology. These are each discussed next.

Proposed active learning workflow for NILM

The proposed AL workflow follows the same steps as described in Section 2.3, Algorithm 1. As shown in Figure 3.1, the dataset is divided into an initial and very small training set ($\boldsymbol{D}_{\text{train}}$), a query pool ($\boldsymbol{D}_{\text{pool}}$) and a test set. Samples from the query pool are considered unlabelled and comprise a representative set of typical on/off samples. A deep learning NILM algorithm is first trained using $\boldsymbol{D}_{\text{train}}$. After the initial training, the obtained model makes predictions on the data from the query pool. The model uses an acquisition function to choose which samples from the query pool should be used for further learning (the set of chosen samples is denoted as $\boldsymbol{Q}$). Having estimated confidence of predictions on data from the query pool, the algorithm queries samples that it was most uncertain about, i.e., the samples that would improve the performance of the model the most, by asking for their corresponding labels. Then these queried samples and their corresponding labels are added to the training data set and they are removed from the query pool. After this step, in the next iteration, the model is trained again with the extended data set that includes newly queried samples. New predictions are made for the

samples left in the query pool, and samples that are chosen for querying are added to the training set and removed from the query pool, and so on. This procedure is repeated until all the samples are queried or a stopping criterion met. The stopping criterion can be, for example, the number of queried samples in total, or the estimated achieved accuracy.

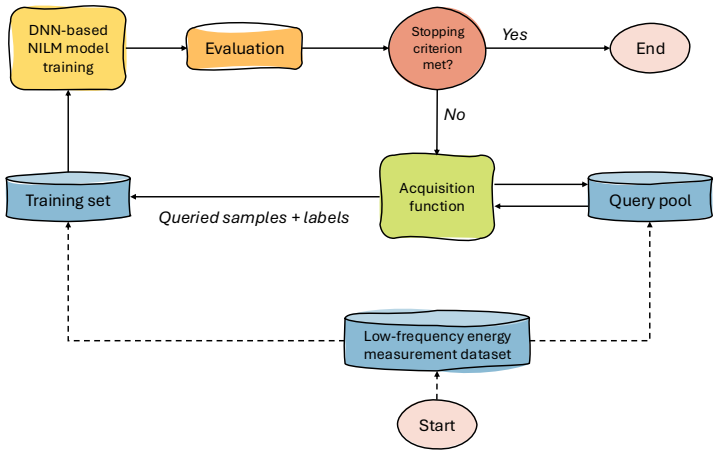


Figure 3.1: The workflow of the proposed active learning framework for model-based low frequency NILM.

Acquisition functions

Acquisition functions are used to choose which samples to query. The goal is to trade-off between the achieved accuracy and the number of queried samples. The acquisition functions are mainly based on estimated model uncertainty, which is assessed through its output, but other approaches are used as well – for example, the distance of a sample from the other available samples, or a combination of the former two methods. In the following the acquisition functions considered are described, adapted here to the low-frequency DNN-based NILM problem.

Uncertainty sampling - least confidence A classification algorithm returns a vector consisting of probabilities of the input samples belonging to

each of the classes present in the data set. This vector can be used to assess the confidence of the algorithm in making its prediction by looking at the probability of the predicted class (the highest probability in the vector). If this value is close to 1, then the model is confident about its prediction. Otherwise, i.e., if none of the class probabilities is significantly larger than others, then the algorithm is not confident in its prediction. That is, for each sample, the highest prediction probability among all the classes can be taken as a measure of confidence.

Since the DNN-based model used for NILM usually process multiple points at a time, returning predictions over a target field, a pooling function that aggregates the model's per-sample probability outputs into a single confidence value for the entire field is defined. In particular, since the state of the whole target field is considered to be on if it contains at least one sample labeled as on-state (i.e., appliance is on at any time inside the target field, and the length of activation does not matter), the maximum probability value of the target field (the maximum among all samples in the field) is used as a confidence measure. If the maximum prediction for the target field is above the decision threshold, the target field is considered positive, and negative otherwise, and the distance from the threshold tells about the model's confidence. This pooling function is used in pool-based or stream-based sampling fashion, as described next.

- Pool-based sampling
  In pool-based sampling, the algorithm makes predictions on the whole query pool (all samples from the pool have to be evaluated), and then a fixed number of predictions that have the lowest confidence values for the predicted class are queried, expertly labelled and added to the training set. The other samples remain in the pool for querying in the next iteration. An example of pool-based sampling is shown in Figure 3.2a with query pool set $\boldsymbol{D}_{\text{pool}}$ of ten samples and four samples $\boldsymbol{Q}$ chosen for query. The numbers represent the output of the model, which can be treated as the certainty of the model (its confidence) when making predictions for each sample from the query pool.

- Stream-based sampling
  In stream-based sampling, samples arrive one by one in a sequence. The decision whether a sample should be queried is made by comparing the probability of the predicted class with a predefined threshold – if the

probability is lower than the threshold, the sample is queried and added to the training set, otherwise, it remains in the pool for querying in the next iteration. An example of stream-based sampling from the same query pool $\boldsymbol{D}_{\text{pool}}$, with threshold $T = 0.9$ [1] and four samples chosen for query $\boldsymbol{Q}$ is shown in Figure 3.2b. If there are less samples than a predefined number of samples whose values fall below the threshold (four in the example in Figure 3.2b, the AL process stops, meaning that the model reached high confidence for most the samples.

BatchBALD Deep learning models typically process a batch of input samples at a time. When using pool- and stream- based uncertainty sampling described above, similarity between chosen samples is not taken into account. If the model is uncertain about one sample, the chances are high that it will be uncertain about other very similar samples (likely to be from the same appliance). That can lead to high redundancy in the chosen samples for querying. In order for learning to progress faster, choosing more diverse batches is necessary.

The BatchBALD [16] acquisition function searches for the optimal batch of samples among all available samples using a greedy approach, based on the joint mutual information between the current batch of samples and the model parameters. In this case, the DNN model needs to be Bayesian, which means that its weights are probability distributions instead of single values. This allows estimating model uncertainty based on the variance in the outputs of multiple runs of a model - the greater the variance, the greater the uncertainty of the model, and vice versa. The score of a batch of samples is calculated according to:

$$\begin{aligned} a_{\text{BatchBALD}}(\{\boldsymbol{x}_1 ... \boldsymbol{x}_{\text{b}}\}, p(\boldsymbol{\omega} \mid \boldsymbol{D}_{\text{train}})) &= I(\boldsymbol{y}_{1:\text{b}}; \boldsymbol{\omega} \mid \boldsymbol{x}_{1:\text{b}}, \boldsymbol{D}_{\text{train}}) \\ &= H(\boldsymbol{y}_{1:\text{b}} \mid \boldsymbol{x}_{1:\text{b}}, \boldsymbol{D}_{\text{train}}) - E_{\text{p}(\boldsymbol{\omega} \mid \boldsymbol{D}_{\text{train}})} H(\boldsymbol{y}_{1:\text{b}} \mid \boldsymbol{x}_{1:\text{b}}, \boldsymbol{\omega}, \boldsymbol{D}_{\text{train}}), \end{aligned} \tag{3.1}$$

where, $\boldsymbol{x}_{1:\text{b}}$ is a batch of b samples drawn from the query pool $\boldsymbol{D}_{\text{pool}}$, $\boldsymbol{y}_{1:\text{b}}$ is the corresponding batch of model predictions, and $\boldsymbol{\omega}$ denote the DNN model

[1] The threshold value is not derived from domain-specific knowledge; it is a methodological hyper-parameter that controls the informativeness of the queried samples. There is not a single correct value to be used; please see Figure 3.8 in Section 3.1.3 under Experiment 3 results for a sensitivity analysis.

parameters. I stands for mutual information, H entropy, E mathematical expectation, and p probability density function.

Bayesian approximation for a standard DNN model can be made using the Monte Carlo (MC) dropout technique [82]. Dropout layers are added to the neural network, and multiple stochastic forward passes are simply collected and averaged. The diversity of prediction probabilities of different forward passes reveals how confident the model is about the sample – the higher the variance the lower the confidence. Importantly, the neural network itself remains unchanged. An example of a batch of samples chosen by the BatchBALD algorithm is illustrated in Figure 3.2c. Note that a batch containing a sample with confidence value of 1 can be selected to be queried, if the diversity of the model output is high among the results of different forward passes.

Random sampling Random sampling, or random query strategy, is the case when a number of samples to be queried is randomly chosen from the query pool - there is no special rule for selecting them, and the model's output for the samples from the query pool is not considered when drawing samples from the pool. This strategy is used as a baseline strategy, and all other strategies which include computing informativeness of samples from the query pool are expected to exceed the prediction performance of the random sampling strategy.

Low-frequency NILM algorithm

For demonstration purposes, the WaveNet-based NILM approach of [43] is selected, which is highlighted [24] as one of the best performing algorithms for low-frequency NILM. A separate model is created for disaggregating each appliance, which facilitates transferability. One of the model's major benefits is that it has a large field of view. It produces concatenated and processed outputs from multiple layers in the network, each with different fields of view, enabling this model to recognise patterns at multiple scales. Since duration of active use times of loads can vary significantly, this feature is favorable. The algorithm performs binary classification in a sequence-to-sequence fashion - that is, it slides a window of input aggregate energy consumption measurements to predict whether an appliance is turned on or off at each point of the sliding window.

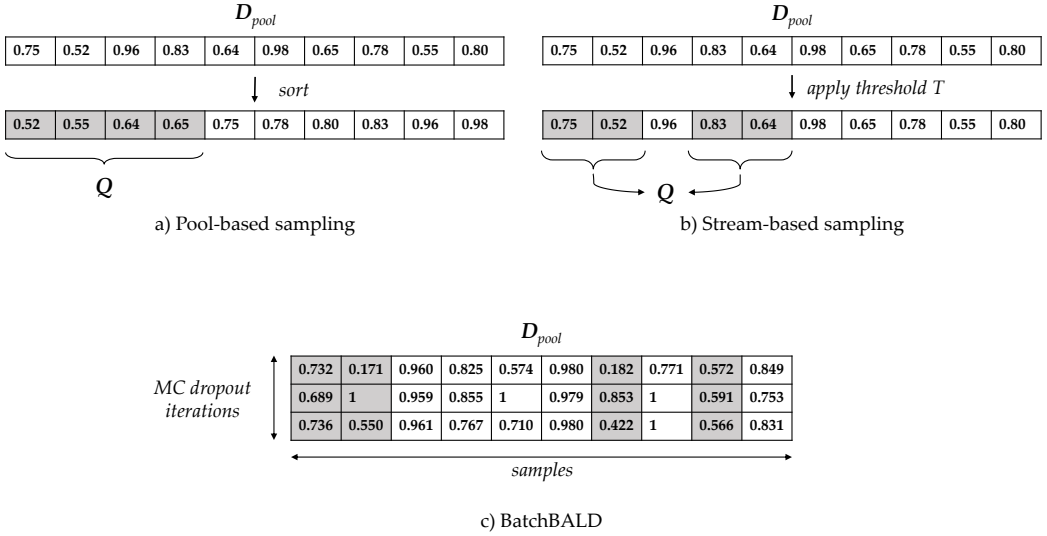


Figure 3.2: Pool-based uncertainty (a), stream-based uncertainty (b) and BatchBALD (c) sampling strategy examples. Each number represents the predicted class probability - certainty of the model when making prediction for each sample in $\boldsymbol{D}_{\text{pool}}$. In this example, four samples are queried per one AL iteration. The used threshold for stream-based uncertainty is T=0.9. The used number of MC dropout iterations (stochastic forward passes) for BatchBALD is 3.

3.1.2 Experimental setup: Dataset, Evaluation metrics and parameter selection

This section provides descriptions of the dataset, evaluation metrics and parameter settings used for demonstration of the proposed methodology. Three experiments are designed to explore the key contributions of this paper. This is followed by the evaluation methodology used to assess the performance of the proposed AL approach.

Dataset

A publicly available electrical load measurement dataset - REFIT [52], as described in Section 2.7.1 was used to showcase the AL methodology in this study.

Appliance models for which AL is developed are kettle, microwave, toaster and dishwasher, due to their high frequency of use, high consumption, and their presence in most houses. As there is imbalance in the on- and off-time for the 4 appliances chosen, data balancing is performed when training, i.e., the same amount of on and off samples is included when generating one batch of data samples to mitigate bias.

Experiments

1. In the first experiment, we assess whether the AL approach can be successfully applied to model-based low-frequency NILM when training and testing domains are the same (i.e., the same house is used for training and testing, albeit with different train, query pool and test sets). Practically, in this scenario, only a small set of labelled measurements is available for initial training of the model. For example, this can be achieved using time-diaries for a short period of time, where householders will keep a time-of-use record of their appliances. During the inference-making process, labelling of queried samples can be achieved as follows: via a domain expert and/or the householder will occasionally be asked to confirm when a particular appliance was run, e.g., via an app.

2. In the second experiment, we test whether AL can enhance the performance of the model when transferred to a new, unseen house. Thus, in practice, time diaries are not needed, since initial training is performed on a publicly accessible dataset. As in Experiment 1, a human will be asked occasionally to label the selected samples from the query pool. We use the data from several houses (excluding the test house) for the initial training set, and the data from the test house for query pool and testing set. Since after each AL iteration, the model is fully retrained using the initial training samples plus all the samples that were queried, the initial training set has to be kept small. Hence, only few REFIT houses are used for training (see Table 3.1).

3. In the third experiment, we use a large pre-training dataset comprising all REFIT houses containing appliances of interest (excluding the test house), instead of a small set of houses as in Experiment 2. When

such a large pre-training dataset is used, it is infeasible to perform full retrain of the model after each AL iteration. Instead, in this experiment, we use AL together with incremental learning [83] to explore if a larger pre-training dataset combined with fine-tuning the model with the samples queried from the unseen house gives better results than using a smaller pre-training dataset and fully retraining the model after each AL iteration. It is important to note that complexity of fine-tuning approach does not depend on the size of pre-training dataset, so it can be arbitrarily large - only newly labelled samples are used when fine-tuning, which is not the case with the full-retrain approach of Experiment 2. In this experiment we also test different settings of AL hyper-parameters - the number of samples queried for a complete AL iteration for pool- and stream-based uncertainty acquisition, and the confidence threshold value for stream-based acquisition function.

In all these experiments various acquisition functions are used and their effectiveness is evaluated. The random query strategy is always used as a baseline.

Parameters

Houses selected for pre-training for each appliance in the second experiment, exploring transferability, are shown in Table 3.1. The choice was made following the example of [44], and by calculating noise-aggregate ratio (NAR) for all houses, so that there are houses with low, middle, and high NAR present in the pre-training data set, given by:

$$NAR = \frac{\sum_{t=1}^{T} \mid y_t - \sum_{i=1}^{M} x_t^{(i)} \mid}{\sum_{t=1}^{T} y_t}. \tag{3.2}$$

Here, y_t denotes the total aggregate energy consumption at time instant t, $x_t^{(i)}$ is the consumption of appliance i, T is the monitoring time period, and M denotes the number of known appliances in the house.

REFIT House 2 is chosen for evaluation due to the fact that it is commonly used for testing in NILM literature - [44, 56], hence it is suitable for validation and benchmarking. In addition, it contains all the appliances of interest, and has a mid-range NAR of 0.67.

All the parameters used for the training of the DNN, as well as in the AL loop, are shown in Table 3.2. The parameters are kept the same as in [43] or are obtained heuristically using the training set. In particular, the input window lengths for kettle and microwave are set to $2^7 - 1$ samples and for dishwasher to $2^{10} - 1$, based on the results reported in [43]. The same window length is set for toaster, since it has similar operation time as kettle and microwave. Target field size of 100 samples is selected as the the best performing in [43]. Training is limited to 20 epochs maximum, because of numerous re-training required during the iterative AL process, and early stopping with patience of 5 epochs is introduced to prevent overfitting. The fine-tuning learning rate is set an order of magnitude lower than the original learning rate used for pre-training, because the weights are already adjusted during pre-training, and although they are tuned, they should not be impacted significantly. In Experiment 3, all trainable network layers are fine-tuned.

The number of samples that are queried for one AL iteration is kept the same as the batch size used in the training process. Data from the target, evaluation house is split into training set (for Experiment 1), query pool and test set so that each set is a representative set of typical on/off samples from the target house. The initial training set size in Experiment 1 is set to only 2^{13} samples, based on the practical assumption that only a small labelled dataset is available (via a small time-diary); a small initial training set also makes the AL process feasible, since the initial training set plus queried samples are all used for model training at each AL iteration. The query pool size is set to 2^{16} samples, to be reasonably larger than the initial training set - to keep the ratio of the labelled and unlabelled number of samples low, and to allow the model to have a variety of samples to choose from, compared to the initial training set. For the BatchBALD acquisition function, the query pool is subsampled to 2^{12} samples, because of the computational demands of the algorithm. The maximum number of queried samples is set to 25% of the whole query pool (i.e., 2^{14}), since this number is sufficient for the performance to stop increasing rapidly (as shown in results, Section 3.1.3), and to keep the time needed for conducting experiments reasonably short. Only for BatchBALD acquisition function, it is set to the whole sub-sampled query pool, considering its size (i.e., 2^{12}).

The confidence threshold for stream-based uncertainty acquisition function is set to 0.9, except for microwave in Experiment 1 and toaster in Experiment 2 it is increased to 0.95, because all the predicted class probabilities

are above 0.9 at the beginning of the AL process, which causes the process to stop without querying any samples. The number of Monte Carlo (MC) dropout iterations that are used in BatchBALD acquisition function is set to 5, which is enough to get a sense of the consistency of model outputs through multiple stochastic forward passes [84, 82].

The performance of the deep learning NILM algorithm is evaluated using F_1 score (Equation 2.4).

AL performance is usually presented as a curve showing model accuracy against the number of labelling iterations, i.e., the number of samples queried and labelled. If a point with no labelling effort (i.e., iteration 0; 0 labelled sata samples), and the maximum possible model performance (i.e., F_1-score equal to 1) is considered as an "ideal" point, then the optimal point of the AL process can be calculated as the point with minimum Euclidean distance from the ideal point:

$$dist = \sqrt{(1 - F_1)^2 + \left(\frac{|\boldsymbol{D}_{\text{ft}}|}{|\boldsymbol{D}_{\text{pool}}|}\right)^2} \tag{3.3}$$

The improvement w.r.t the initial model performance - $F_{1\text{ initial}}$, when none of the samples from the query pool are labelled and added to training, and a gap to the heuristic bound performance - $F_{1\text{ bound}}$, achieved when the whole query pool is labelled, are calculated according to the following equations:

$$improvement = \frac{F_1 - F_{1\text{ initial}}}{F_{1\text{ initial}}}, \tag{3.4}$$

$$gap = \frac{F_{1\text{ bound}} - F_1}{F_{1\text{ bound}}}. \tag{3.5}$$

It is expected that by adding new samples to the initial training set, the performance will improve. However, the improvement could be negative if the performance drops, due to, for example, adding non-informative samples to the training set from the query pool. On the other hand, the results are expected to be worse compared to the heuristic bound $F_{1\text{ bound}}$, but the results could exceed this bound, due to, for example, overfitting the model with a very large training dataset, which would lead to the gap being negative.

Specifications of the PC used for experiments are: Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz, 32GB RAM, and a NVIDIA TITAN Xp GPU.

Table 3.1: On-state power threshold in [W] and training houses in Experiment 2 for each target appliance.

Appliance	Training houses	NAR	On power threshold [W]
Kettle	House 6	0.69	2000
	House 8	0.78	
	House 17	0.58	
Microwave	House 6	0.69	200
	House 8	0.78	
	House 17	0.58	
Toaster	House 6	0.69	50
	House 7	0.58	
	House 8	0.78	
Dishwasher	House 3	0.56	10
	House 6	0.69	
	House 9	0.61	

3.1.3 Results & Discussion

In this section we present results from each of the three experiments described in Subsection 3.1.2. We discuss the performance of AL, transfer learning of DNN-based NILM models with AL, retraining the whole model using the entire training dataset or only fine-tuning using the new labelled samples after each iteration, as well as the effect of different acquisition functions on performance and transferability in a realistic scenario - using real, dynamic household measurements. In addition, we discuss sensitivity to AL hyperparameters. All the curves in the plots are smoothed using Savitsky-Golay filter of order 3 and window length 11.

Experiment 1 Results

The results from the first experiment - demonstrating that AL can be successfully applied to model-based low-frequency NILM by taking data from a single REFIT house, House 2, for the initial training, query pool and test sets, are shown in Figure 3.3. The horizontal axis shows the percentage of samples from the query pool that are labelled, and the vertical axis shows F_1 score achieved by the model. The red dotted line reports $F_{1\ \text{bound}}$, when model is trained on the initial training data set and the whole query pool

Table 3.2: Model training and active learning hyper-parameters. 1 sample = 1 window.

Parameter		Value
Input window size	kettle, microwave, toaster	$2^7 - 1$
	dishwasher	$2^{10} - 1$
Target field		100
Batch size		2^7
Number of maximum epochs		20
Early stopping patience (epochs)		5
Learning rate		10^{-3}
Fine-tuning learning rate		10^{-4}
Number of samples queried per active learning iteration		2^7
Initial training set size for Experiment 1 (samples)		2^{13}
Query pool size (samples)	BatchBALD	2^{12}
	other query strategies	2^{16}
Number of maximum queried samples		2^{14}
Confidence threshold	Exp. 1 - microwave & Exp. 2 - toaster	0.95
	all other experiments	0.9
Number of MC dropout iterations		5

(100%) together. Those performance bounds are inline with those reported in [43]. The black dotted line shows the initial F_1 score obtained by using the initial training set only.

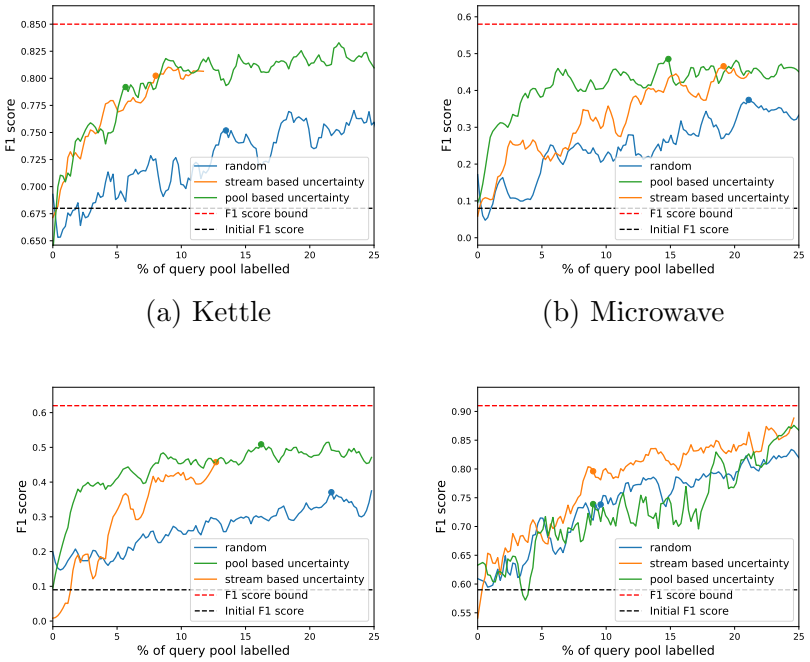
Note that the experiments were not run until the whole query pool is added to the training data set, but were stopped after 25% of the query pool is added, so the plots show the performance up to that point. For the stream-based uncertainty acquisition function, the AL process can stop earlier if the stopping criterion is met, i.e., there are insufficient samples with probability of the predicted class below the threshold to form a batch.

The optimal points calculated according to (3.3) are also marked in the AL curves for each appliance and query strategy explored in corresponding colours in Figure 3.3, showing the best trade-off between labelling effort and accuracy achieved. As expected, the performance of all methods increases with the number of samples added to the training set, and it increases faster for the pool- and stream-based uncertainty acquisition functions than it does for random sampling. Therefore, AL gives promising results for the training models to disaggregate kettle, microwave and toaster.

It can be seen from Figure 3.3, that pool-based and stream-based sampling achieve the optimal performance-complexity point very early (after as little as 5% for kettle and 15% for toaster and microwave, of labelled samples added to the training set), and much before the random sampling baseline except for dishwasher.

For dishwasher there is an increase in performance with samples being labelled, mainly in the range between 1% and 17% of the query pool samples labelled; the increase of random sampling is the same as that of the pool-based strategy, implying that the contribution of all samples in the query pool is similar, or that the pool-based query strategy cannot identify the most informative samples. The stream-based sampling, however, consistently outperforms the other two methods.

Table 3.3 shows the portion of the query pool that needs to be added to the training set so that the model exceeds 90% of the heuristic bound performance. If 90% was not achieved, the maximum F_1 score and the corresponding portion of query pool are shown. It is worth noticing that with only up to 20% of the query pool samples being labelled and added to the training set, the performance is close to the bound for all appliances, which indicates that the labelling effort could be reduced by as much as 80%. The smallest labelling effort is required for kettle, whose performance is very good to start with, and is of short duration (hence, with a small number of queried samples,



(a) Kettle

(b) Microwave

(c) Toaster

(d) Dishwasher

Figure 3.3: Experiment 1: Models trained and tested on REFIT House 2 for kettle (a), microwave (b), toaster (c) and dishwasher (d). The red broken line shows the F_1 score bound obtained by using the entire query pool (100%) for training. The dots represent the optimal points obtained using (3.3). The black broken line is the result obtained with initial training only (0% query pool labelled).

many activations can be processed). On the other hand, the most labelling effort is required for toaster and microwave, due to the fact that the model does not disaggregate these two appliances well, as can be seen from the final performance bound, which is around 0.6 for both these appliances. This can also be due to the fact that microwave and toaster have a more statistically complex load profile compared to kettle and are used with different settings, hence more samples are needed to capture the statistics. Interestingly, both pool-based and stream-based sampling achieve similar performance, indicating that off-line labelling is not needed and samples can therefore be labelled as they arrive.

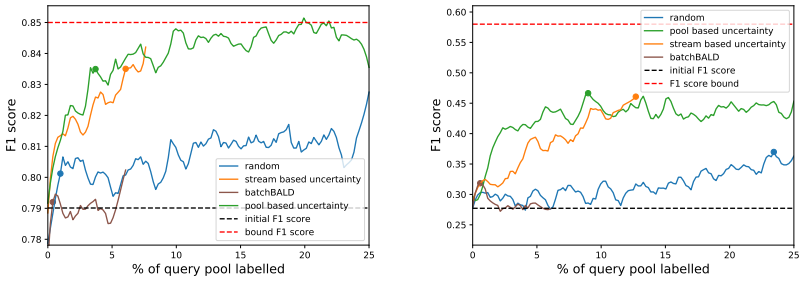
Table 3.3: Experiment 1: Labelling effort, i.e., % of the labelled query pool samples, $|\boldsymbol{Q}|$, needed to exceed 90% of the bound F_1 score (if possible). The bound F1 corresponds to the results when the entire query set (100%) is used for training.

		Kettle	Microwave	Toaster	Dishwasher
Pool-based	$\lvert \boldsymbol{Q} \rvert$ / $\lvert \boldsymbol{Q}_{\text{pool}} \rvert$	1.6%	10.2%	18.5%	15.8%
	F_1 / $F_{1\text{ bound}}$	92%	90%	90%	90%
Stream-based	$\lvert \boldsymbol{Q} \rvert$ / $\lvert \boldsymbol{Q}_{\text{pool}} \rvert$	1.6%	19.33%	12.1%	8.4%
	F_1 / $F_{1\text{ bound}}$	90%	91%	82%	90%

Experiment 2 Results

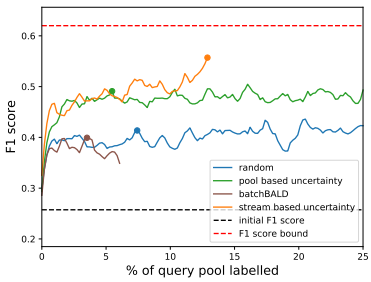
The results of Experiment 2 are shown in Figure 3.4. A pre-trained model is transferred to unseen REFIT House 2, and the samples from this house are gradually labelled and added to the training set. The black dotted line represents the disaggregation performance of the pre-trained model on House 2 data without any data from that house added to the training set (0% of the query pool sampled labelled), i.e., before any adaptation to the new environment. The red dotted line reports the heuristic bound F_1 score as in Experiment 1. Even though the query pool for BatchBALD acquisition function is sub-sampled from the original larger pool, curves are shown with respect to the larger pool, to line up the number of queried samples with other acquisition functions.

As can be seen from the plots in Figure 3.4, the proposed AL approach yields promising results for all four appliances tested. As expected, strategically selecting the samples to query significantly improves the performance w.r.t random sampling. Pool- and stream-based uncertainty acquisition functions perform similarly, with pool-based being slightly better for kettle and microwave, and stream-based being slightly better for dishwasher until it reaches high confidence for all samples belonging to the pool. This can also be observed by the optimal points that are reached very early (after only 5-10% samples labelled). The performance of dishwasher has the steepest increase over a number of iterations. This is expected due to dynamic nature of dishwasher loads within the house - newly added samples provide new information due to variation in dishwasher power patterns over different runs. This is less pronounced with kettle and microwave since newly added samples after 5-10% of query pool samples being added do not enlarge anymore

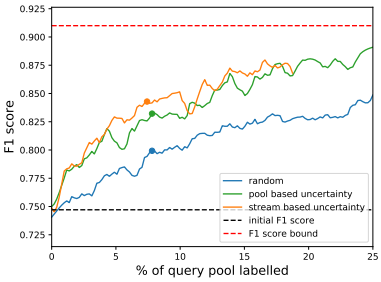


(a) Kettle

(b) Microwave



(c) Toaster



(d) Dishwasher

Figure 3.4: Experiment 2: Models pre-trained with small dataset transferred to REFIT House 2 for kettle (a), microwave (b), toaster (c) and dishwasher (d). Full retrain of the model is performed in each AL iteration. The red broken line shows the F_1 score bound as per Experiment 1. The broken black line shows the initial F_1 score obtained using pre-training set only. The dots represent the optimal points obtained using (3.3).

the informativeness of the training pool. Regarding the toaster, there is a huge jump immediately when fine-tuning is performed due to a large difference between the toaster signature in the target domain (House 2) and those available in the training set. However, after that, the newly added samples do not improve the performance anymore, which can be attributed to the fact that disaggregating toaster is in general very challenging and the results have already come closer to the bound in Figure 3.3.

The BatchBALD acquisition function performs similarly to the random acquisition function, which can be explained by the very limited size of the query pool. The BatchBALD acquisition function is very computationally

expensive and could not handle a large query pool due to memory constraints. It is not used for dishwasher due to the extremely small query pool size, and hence observed lack of improvement beyond the initial training.

Table 3.4 shows the improvement w.r.t the initial performance and the gap to the heuristic bound as defined in Equations (3.4) and (3.5). The best results are shown (maximum performance) within the first 25% of samples added to training, as well as the results with the optimum trade-off points. The results show a high level of improvement for all appliances, bearing in mind that a much higher improvement is desired for lower-performing initial models, i.e., microwave and toaster, since the initial results for kettle and dishwasher were already high. A very small gap for kettle, dishwasher and toaster with pool- and stream-based sampling indicates that there is very little room for improving querying strategies. The optimal trade-off points are generally close to the maximum performance.

Table 3.4: Experiment 2: The improvement of the initial performance of the NILM model transferred to a new house using AL when labelling at most 25% of the query pool, and the gap to the heuristic bound. The results are given for the optimal trade-off point as well as for the best performance.

		Kettle	Microwave	Toaster	Dishwasher
		Maximum performance			
Pool-based	Improvement	8.68%	79.23%	104.00%	19.42%
	Gap	-1.02%	14.34%	15.31%	1.95%
Stream-based	Improvement	6.77%	73.63%	122.11%	18.59%
	Gap	0.75%	17.02%	10.14%	2.64%
BatchBALD	Improvement	1.70%	22.51%	72.38%	-
	Gap	5.47%	41.45%	28.43%	-
		Optimal trade-off points			
Pool-based	Improvement	7.85%	76.26%	97.71%	13.63%
	Gap	-0.25%	15.76%	17.92%	6.71%
Stream-based	Improvement	6.77%	73.63%	122.11%	13.17%
	Gap	0.75%	17.02%	7.79%	7.09%

Table 3.5 shows the comparison of F_1 score when initially training the model using data from the same house where the model will be deployed (no transfer), and when a pre-trained model, trained with already available data from multiple houses is transferred to the new house. One can see that both

sampling strategies show very small drop in performance when transferred to a new target domain, indicating very fast adaptation due to effectively using the query pool.

Note that the models pre-trained with data from multiple houses can perform better than models trained and tested using data from the same house. This is due to the fact that, as per Experiment 1 settings, initial training set is of very limited size when training and testing with data from the same house, as in a practical scenario, those data will be obtained from time-diaries kept by householders. On the other hand, in Experiment 2, larger amount of data from multiple houses, from an already available, public dataset containing submeter measurements is used, which offers a better variety of data samples for the model to learn.

Table 3.5: Comparison of the transfer learning results (Experiment 2) and no-transfer learning (Experiment 1) in terms of the maximum F_1 score achieved when labelling at most 25% of query pool. The best results are shown in bold.

		Kettle	Microwave	Toaster	Dishwasher
		Maximum performance			
Pool-based	No-transfer	0.8511	**0.5756**	0.5626	0.8860
	Transfer	**0.8587**	0.4968	0.5251	**0.8922**
Stream-based	No-transfer	0.8241	0.5254	0.5142	0.8897
	Transfer	0.8436	0.4813	**0.5717**	0.8860
		Optimal trade-off points			
Pool-based	No-transfer	0.8217	**0.5591**	0.5501	0.8046
	Transfer	**0.8521**	0.4886	0.5089	**0.8489**
Stream-based	No-transfer	0.8291	0.5254	0.5142	0.8324
	Transfer	0.8436	0.4813	**0.5717**	0.8455

Considering the presented results of this experiment, it can be concluded that AL can be used to effectively enhance the performance of pre-trained AL models when transferred to a new environment, whose appliance profiles (e.g., toaster) are statistically different. Similarly to Experiment 1, stream based sampling shows no performance loss compared to pool based sampling, thereby indicating that online learning is possible.

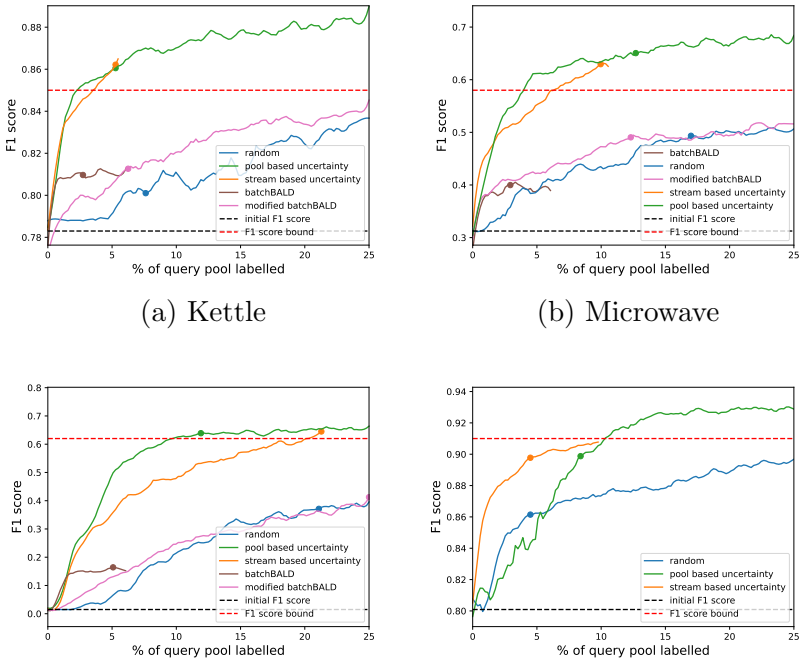
Experiment 3 Results

In Experiment 2, after each iteration, when new samples are added to the training set, the entire model is retrained, as is commonly performed in the AL literature. However, due to these frequent re-training process, the initial training set has to be kept very small, and therefore the execution time to obtain improvements is high. To attempt to mitigate the aforementioned problem, in Experiment 3, we do not retrain the entire model after each iteration, which enables us to increase the size of the initial training set. The results of this experiment - i.e., transfer of a DNN-based NILM model to a new house with a large pre-training dataset and fine-tuning are shown in Figure 3.5 and Table 3.6.

Table 3.6: Experiment 3: F_1 score achieved by the NILM model transferred to a new house using the large pre-training dataset and the fine-tuning approach to AL when labelling at most 25% of query pool.

		Kettle	Microwave	Toaster	Dishwasher
		Maximum performance			
Pool-based	Improvement	12.79%	122.69%	4475.845%	16.84%
	Gap	-4.58%	-19.98%	-9.97%	-2.56%
Stream-based	Improvement	9.71%	103.55%	4226.17%	13.95%
	Gap	-1.72%	-9.67%	-3.97%	-0.02%
BatchBALD	Improvement	3.58%	35.23%	1090.60%	-
	Gap	3.96%	27.14%	71.39%	-
Modified BatchBALD	Improvement	7.26%	71.55%	2752.35%	-
	Gap	0.55%	7.57%	31.45%	-
		Optimal trade-off points			
Pool-based	Improvement	9.62%	116.48%	4243.62%	12.97%
	Gap	-1.64%	-16.64%	-4.39%	0.84%
Stream-based	Improvement	9.71%	103.55%	4226.17%	12.47%
	Gap	-1.72%	-9.67%	-3.97%	1.27%
Modified BatchBALD	Improvement	3.88%	62.30%	2752.35%	-
	Gap	3.68%	12.55%	31.45%	-

It can be seen from Figure 3.5, that the AL process in this experiment is more stable - AL curves do not deviate with fine-tuning, especially in the beginning of the process, which is expected since the models are not fully retrained. The optimal trade-off points are again achieved early, with



(a) Kettle (b) Microwave

(c) Toaster (d) Dishwasher

Figure 3.5: Experiment 3: Models pre-trained with large datasets transferred to REFIT House 2 for kettle (a), microwave (b), toaster (c) and dishwasher (d). Fine-tuning of the model is performed in each AL iteration without retraining. The red broken line shows the F_1 score bound as in Experiment 1. The broken black line shows the initial F_1 score obtained using pre-training set only. The dots represent the optimal points obtained using (3.3).

only 5-15% of added labelled samples, and as observed in previous experiments before, pool-based and stream-based uncertainty sampling lead to similar performance.

In Table 3.6, gap values are negative both for maximum performance (for all appliances) and optimal points (for all appliances except dishwasher, where it is still very small) when using pool- and stream-based sampling strategies, meaning that bound performance is exceeded, implying that it is worth to use large pre-training datasets and fine-tuning approach.

For toaster, the pre-trained model performs poorly in the new house, but despite that, a higher F_1 score is achieved compared to Experiment 2.

Poor initial performance is attributed to the statistical diversity in toaster models, and the fact the House 2 toaster model, and hence load profile, is not available in other houses; however, the AL approach with fine-tuning overcomes this problem, as shown in Figure 3.5c and negative gap values in Table 3.6.

Due to the high computational demands of retraining, BatchBALD in Experiment 2 could handle only a limited number of samples from the query pool. In this experiment, since re-training is not performed after each label is added, but only fine-tuning, we adapt BatchBALD such that the query pool updates each time a batch of samples is drawn out of it and newly arrived samples are put in the pool to replace the drawn ones. Thus, this could be considered as a hybrid of a pool- and stream-based acquisition and is referred to modified BatchBALD.

The proposed modified BatchBALD method with the introduced adaptation performs better than random sampling for kettle and microwave, compared to the bound performance. In general, BatchBALD performs worse than pool- and stream-based uncertainty sampling, which can be explained by the fact that all samples in the query pool are not highly correlated and it is sufficient to look at their importance and not mutual correlation.

A comparison of full retrain (Experiment 2) and fine-tuning (Experiment 3) in terms of F_1 score is presented in Table 3.7. Looking at the plots in Figure 3.5, and at Table 3.7, it can be observed that the performance of the model that is pre-trained using a very large dataset and fine-tuned with queried samples reaches higher F_1 score for all appliances tested than the model that is pre-trained using a smaller dataset and fully retrained at each iteration (i.e, Experiment2).

Using models pre-trained with large datasets and fine-tuning, instead of full retrain, yields the best results among all 3 experiments, with an important benefit that should not be neglected - a significant decrease in time needed for completing the AL process. An insight in speed-up that the fine-tuning approach enables is shown in Figure 3.6 for various sizes of the pre-training dataset, by using the number of samples included in training as an indicator of time needed for training. The speed-up S is computed as a ratio of samples included in the model training with the full retrain approach (pre-training samples + queried samples) denoted as $\mid \boldsymbol{D}_{pre-train} \mid$, and samples included in the model training with the fine-tuning approach (queried samples only, $\mid \boldsymbol{Q} \mid$), according to:

Table 3.7: Comparison of full retrain (Experiment 2) and fine-tuning (Experiment 3) - for each appliance the best F_1 score the model achieved when at most 25% of the query pool is labelled.

		Kettle	Microwave	Toaster	Dishwasher
		Maximum performance			
Pool-based	Full retrain	0.8587	0.4968	0.5251	0.8922
	Fine-tuning	**0.8889**	**0.6959**	**0.6818**	**0.9333**
Stream-based	Full retrain	0.8436	0.4813	0.5571	0.8860
	Fine-tuning	0.8646	0.6361	0.6446	0.9102
BatchBALD	Full retrain	0.8035	0.3396	0.4437	-
	Fine-tuning	0.8163	0.4226	0.1774	-
		Optimal trade-off points			
Pool-based	Full retrain	0.8521	0.4886	0.5089	0.8489
	Fine-tuning	0.8639	**0.6765**	**0.6472**	**0.9024**
Stream-based	Full retrain	0.8436	0.4813	0.5717	0.8455
	Fine-tuning	**0.8646**	0.6361	0.6446	0.8984

$$S = \frac{|\, \boldsymbol{D}_{\text{pre-train}} \,| + |\, \boldsymbol{Q} \,|}{|\, \boldsymbol{Q} \,|}. \tag{3.6}$$

As it can be seen from Figure 3.6, the larger the pre-training dataset, the higher the speed-up of the fine-tuning approach. The fine-tuning approach offers significant time savings, most of which happens in the early AL process, which is when the model's performance increase is most rapid, as per the results of all aforementioned experiments. Moreover, as mentioned before, with fine-tuning, the size of pre-training dataset can be arbitrarily large, since only the queried samples are used during training.

The results of sensitivity analysis regarding the number of samples queried for one iteration for random, pool- and stream-based uncertainty are shown in Figure 3.7. Note that the horizontal axis of the plot shows the percent of the query pool labelled, i.e., the labelling effort. It can be seen from the figure that the performance is not sensitive to the number of queries per iteration.

Results of sensitivity analysis with regards to the confidence threshold used for stream-based uncertainty acquisition function are presented in Figure 3.8. A lower confidence threshold leads to more challenging samples

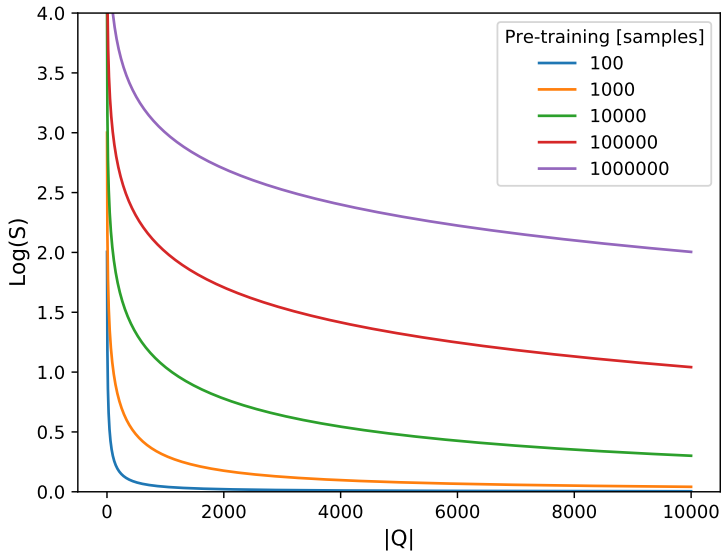
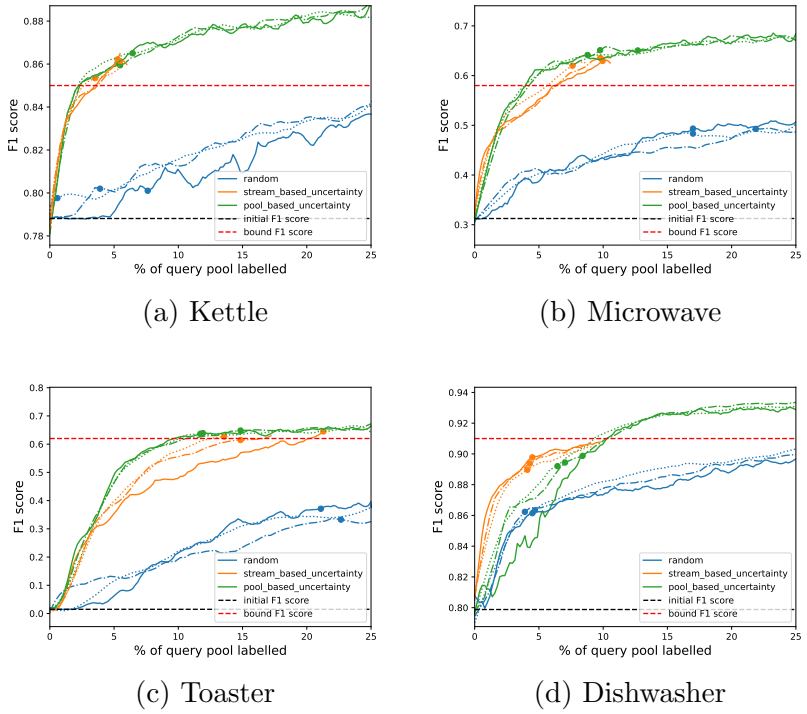


Figure 3.6: The speed-up of fine-tuning compared to the full retrain approach to AL for various sizes of the pre-training dataset (in the number of samples). The horizontal axis shows the number of labelled samples from the query pool.

added to the training set, and hence faster improvement in performance compared to higher thresholds. On the other hand, a higher confidence threshold implies that more samples are going to be considered, so the process runs for longer. For dishwasher, all confidence threshold levels provide equally steep performance increase, which is likely due to a large number of samples with the confidence level below the lowest threshold (0.9), caused by other loads with similar wattage present in the training dataset, for example, dishwasher is often confused with washing machine [27].

Results Summary

- AL can be successfully applied to model-based low-frequency NILM to reduce labelling effort, and to enhance performance of models transferred to new environments.
- Performance of stream-based acquisition function, that can be per-



(a) Kettle (b) Microwave

(c) Toaster (d) Dishwasher

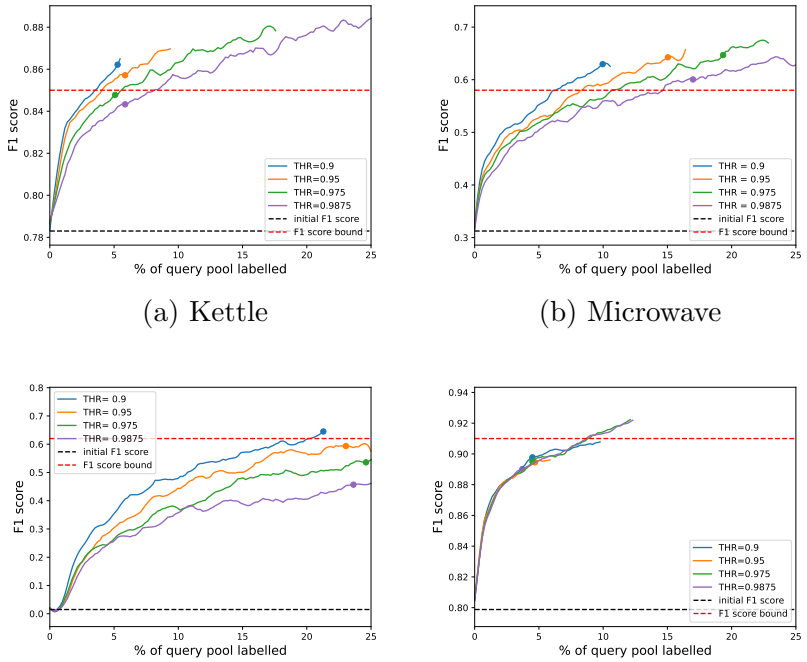
Figure 3.7: Experiment 3 - Sensitivity analysis: Models pre-trained with large datasets transferred to REFIT House 2 for kettle (a), microwave (b), toaster (c) and dishwasher (d). Fine-tuning of the model is performed in each AL iteration with a variable number of samples queried - 128 (solid line), 256 (dash-dotted line) and 384 (dotted line).

formed online, is on par with pool-based one that requires presence of the whole query pool in advance and hence cannot be used online.

- Batch-aware acquisition function (BatchBALD [16]) was inferior to other acquisition functions explored, due to its high computational demands. To mitigate the complexity and low accuracy of the original BatchBALD, a modification of it has been introduced.

- Optimal trade-off between accuracy and labelling effort is achieved with 5-15% of query pool labelled in most of the cases.

- Fine-tuning offers a good trade-off between accuracy and labelling effort



(a) Kettle (b) Microwave

(c) Toaster (d) Dishwasher

Figure 3.8: Experiment 3 - sensitivity analysis: Models pre-trained with large datasets transferred to REFIT House 2 for kettle (a), microwave (b), toaster (c) and dishwasher (d). Fine-tuning of the model is performed in each AL iteration using the stream-based uncertainty acquisition function with different confidence thresholds (THR).

and therefore full retrain at each iteration may not be necessary.

- Performance of AL with pool- and stream-based acquisition functions is not sensitive to the number of samples queried per iteration - same labelling effort yields same performance, but if more samples are queried in one iteration, fewer iterations are required.

- The lower the confidence threshold for stream-based uncertainty acquisition function, the faster the improvement of the model in the beginning of the AL process; the higher the confidence threshold, the longer the process runs.

3.1.4 Summary

In order to take advantage of large scale smart meter rollout and NILM to be deployed widely to get itemized electricity consumption reports for improved energy management, it is important to have a way to adapt NILM algorithms to new houses efficiently, to get best-performing algorithms with as little labelled data as possible. This paper demonstrated the viability of AL to reduce labelling effort, as well as to improve transferability of deep learning models to statistically different and dynamic electrical measurements. Three different experiments were conducted - first, to show that labelling effort can be significantly reduced by using AL and providing labels only for valuable samples; second, to show that the performance of DNN-based NILM models with AL, can be enhanced when transferred to a new environment by labelling reasonably small amount of new samples that are informative; and third, to show that full retrain of deep learning models after each AL iteration may not be necessary - fine-tuning with only newly labelled data from the new environment can produce satisfactory results, offering a good trade-off between performance achieved and computational resources needed.

Different acquisition functions were explored, including pool- and stream-based uncertainty, and batch-aware BatchBALD acquisition function along with a modified BatchBALD to address complexity of the original BatchBALD. Worth noting is that the performance of the stream-based uncertainty, which can be implemented online, was on par with pool-based uncertainty, which requires availability of the whole query pool in advance, and hence cannot be implemented online. BatchBALD acquisition function can consider only small query pool sizes, because of its high computational requirements, and therefore its performance was inferior to other acquisition functions. To overcome this, a modification is introduced to update the query pool in a stream-like fashion, to obtain a hybrid of pool- and stream-based strategy. Though the modified BatchBALD outperformed the original BatchBALD, its performance is still inferior to pool- and stream-based uncertainty strategies. Optimal trade-off between labelling effort and accuracy was discussed - in most of the cases, the optimal point was achieved with 5-15% of query pool labelled, which indicates that labelling effort could be reduced by as much as 85%. Changing number of samples queried per AL iteration offers achieving the same performance in lower number of iterations, but with the same labelling effort. Setting lower threshold for stream-based uncertainty acquisition function provides steeper increase in performance, while setting

higher threshold offers a longer lasting AL process process.

3.2 A weakly supervised active learning framework for non-intrusive load monitoring

In the previous section on AL for the NILM problem, labels provided during the AL process are of the same granularity as recordings of electricity consumption. In practice, this kind of label can be acquired via submetering, i.e., installation of an individual meter for each appliance. When there is no submetering, this high granularity is only possible if a NILM expert is providing labels, because such fine-grained labelling requires knowledge of consumption signatures of different types of appliances, and the ability to recognise them within a noisy recording. On the other hand, weak labels are given per window of the electricity consumption signal, indicating if an appliance is active inside that window. This kind of label can be provided by an end user, through an app, based on the time of use of individual appliances, without requirements for specific knowledge of appliance signatures and their extraction from noisy aggregate recordings.

3.2.1 Methodology

The AL framework used in this study is presented in Figure 3.9. The process follows the same steps as described in Section 2.3. Weak labels are provided by an oracle, and they are used to fine-tune the model, but evaluation is performed using strong labels. Acquisition function used in this study is uncertainty based. Weak level prediction $\hat{w}$ of the model is a vector containing probabilities of each appliance being in an active state inside the input signal window. These values are used to estimate uncertainty of the model. Namely, to finally determine the state of an appliance k, it's soft prediction $\hat{w}_k$ is compared to a threshold β - if it is greater that the threshold, the appliance is considered to be active, and otherwise it is considered to be turned off. The closer the value of $\hat{w}_k$ to β, the more uncertain the model is about the prediction. Therefore, we define uncertainty δ_k of the model about the prediction for appliance k as follows:

$$\delta_k = \begin{cases} \hat{w}_k & \hat{w}_k < \beta \\ 1 - \hat{w}_k & \hat{w}_k \geq \beta \end{cases} \tag{3.7}$$

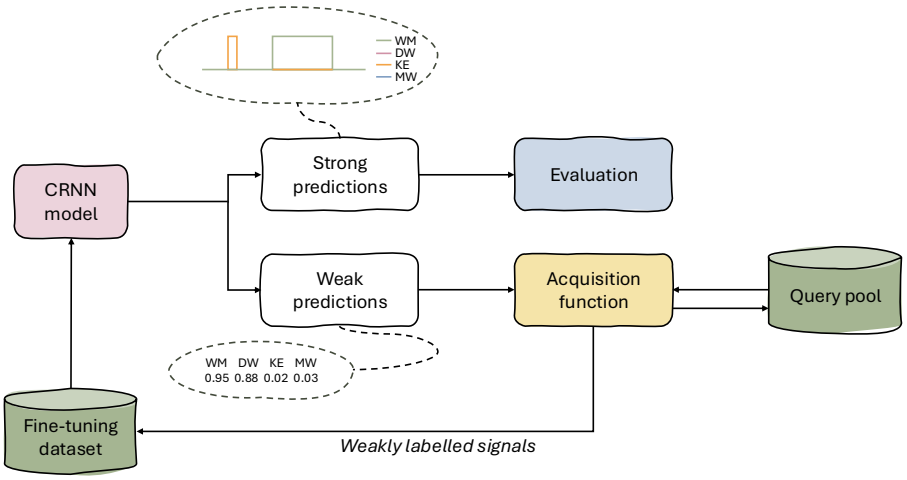


Figure 3.9: Weakly supervised AL scheme.

Since the problem considered in this study is a multi-label classification, where the state of K appliances is predicted by the model at the same time, the following two strategies are considered to estimate the overall uncertainty δ of the model:

- maximizing uncertainty level across all appliances:

$$\delta = \max_k \delta_k \tag{3.8}$$

- averaging uncertainty across all appliances:

$$\delta = \frac{1}{K} \sum_{k=1}^{K} \delta_k \tag{3.9}$$

In each AL iteration, a batch of signal windows with the highest uncertainty level is selected to be weakly labelled and included in the fine-tuning set.

The DNN model used within the framework is a multi-label classification CRNN from [85], proven to perform well when trained with weak labels [85].

The two datasets used in experiments of this study are REFIT and UK-DALE, as described in Section 2.7.1.

Experimental setup

UK-DALE houses 1, 3 and 5 were used for pre-training of the model (refer to [85]), while REFIT houses 2, 4, 5 and 19 are used for fine-tuning during the AL process. Aggregate and appliance-level signals from REFIT are up-sampled uniformly to 1/6 Hz, to align with UK-DALE used for pre-training. Window length is set to 2550 samples, which equals 4.15 hours. The number of signal windows from REFIT dataset per appliance from each house used is shown in Table3.8. The query pool is composed of 30% of REFIT signal windows from each house, and test set of the remaining 70%.

House	KE	MW	WM	DW	Total # of windows
House 2	2.9	-	2.9	2.9	2.9
House 4	12	12	-	-	12
House 5	9.5	-	0.5	0.5	9.5
House 19	13.6	13.6	-	-	13.6

Table 3.8: Number of sample windows for each appliance from each REFIT house (given in thousands).

In this study, two experimental scenarios are considered:

1. Active learning with the CRNN model pre-trained with only weakly labelled data from UK-DALE dataset.

2. Active learning with the CRNN model pre-trained with both weakly and strongly labelled data from UK-DALE dataset.

In Scenario 1, fine-tuning bi-directional and instance layers of the CRNN yielded best results, while in Scenario 2, fine-tuning only the instance layer performed the best. The threshold β used to determine the state of appliances is calculated using the optimal thresholding strategy for each pre-training condition. Adam optimizer is used with a learning rate of 0.002. Batch size used is 64. Metrics used to evaluate the CRNN performance are F_1 and micro-F_1 score (Equations 2.4 and 2.6, respectively).

AL performance is usually presented as a curve showing model accuracy against the number of labelling iterations, i.e., the number of samples queried and labelled. If a point with no labelling effort (i.e., iteration 0; 0 labelled sata samples), and the maximum possible model performance (i.e., F_1-score equal to 1) is considered as an "ideal" point, then the optimal point of the

AL process can be calculated as the point with minimum Euclidean distance from the ideal point (see Eq. 3.3).

The performance of the proposed framework is compared to "No Fine-Tuning" model [85], i.e., the model before adding any weakly labelled data from the target environment, as well as to the "Weak Transfer Learning" model [85], with the whole weakly labelled query pool added. Moreover, the proposed method is compared to the semi-supervised method of [86] based on knowledge distillation, pre-trained with strongly labelled data, and fine-tuned with unlabelled data from the target environment (labelled data from the target domain are considered unavailable). During the AL process, the unlabelled signal windows associated with the highest uncertainty level are selected for fine-tuning.

3.2.2 Results & Discussion

Results of the semi-supervised benchmark method [86] are presented first, in Table 3.9. This is a challenging setting since no labelled data from the target domain are provided - fine-tuning is done with unlabelled data only. Even though some improvement can be achieved compared to the case when no data from target domain is used at all, e.g., in House 4, it is not sufficient, and adding labelled data is advantageous. Thus, results from AL scenarios with weakly labelled data from the target domain are presented next.

Scenario 1 results, with only weakly labelled data available during both pre-training and the AL phase, are presented in Table 3.10. This scenario is challenging due to the fact that the model never sees strong labels, neither during the pre-training, nor during the AL stage. Kettle has a short activation time, which makes it more likely to coincide with other appliances, and therefore it needs a larger number of queries to get enough kettle activations within different aggregates. Similar holds for microwave. Washing machine has a more complicated signature, and therefore does not improve with adding weak labels. Dishwasher, however, has more high-power samples in an activation, which makes it easier to improve with adding weakly labelled data samples. House 2 has the lowest noise-aggreagte ratio (NAR, [87]) of 0.79. House 4 is noisier, with NAR equal to 0.91, hence the starting performance is worse. House 5 has NAR of 0.84, therefore, better starting performance is observed. House 19 has the highest NAR value, 0.93, however, the starting performance is good, indicating similarity of the appliances in House 19 to the ones seen during the pre-training phase.

	Method		KE	MW	WM	DW	micro F_1
H2	No Fine-Tuning		0.55	-	0.48	0.58	0.50
	Unsup. Transfer		0.55	-	0.48	0.58	0.50
	AL(max)	opt.	0.55(13.3%)	-	0.41(6.7%)	0.58(6.7%)	0.50(6.7%)
		best	0.56(80%)	-	0.41(6.7%)	0.58(6.7%)	0.50(6.7%)
	AL(mean)	opt.	0.54(13.3%)	-	0.41(6.7%)	0.58(6.7%)	0.50(6.7%)
		best	0.56(73.3%)	-	0.41(6.7%)	0.58(6.7%)	0.50(6.7%)
H4	No Fine-Tuning		0.42	0.38	-	-	0.39
	Unsup. Transfer		0.44	0.44	-	-	0.44
	AL(max)	opt.	0.44(13.8%)	0.41(10.3%)	-	-	0.42(13.8%)
		best	0.45(20.7%)	0.44(38%)	-	-	0.44(38%)
	AL(mean)	opt.	0.45(1.7%)	0.41(12.1%)	-	-	0.41(12.1%)
		best	0.45(1.7%)	0.44(98.2%)	-	-	0.44(98.2%)
H5	No Fine-Tuning		0.86	-	0.02	0.04	0.05
	Unsup. Transfer		0.86	-	0.02	0.04	0.05
	AL(max)	opt.	0.86(4.3%)	-	0.02(2.2%)	0.04(2.2%)	0.05(2.2%)
		best	0.87(60.9%)	-	0.02(2.2%)	0.04(2.2%)	0.05(2.2%)
	AL(mean)	opt.	0.86(4.3%)	-	0.02(2.2%)	0.04(2.2%)	0.05(2.2%)
		best	0.87(97.8%)	-	0.02(2.2%)	0.04(2.2%)	0.05(2.2%)
H19	No Fine-Tuning		0.82	0.61	-	-	0.69
	Unsup. Transfer		0.82	0.61	-	-	0.69
	AL(max)	opt.	0.82(3.1%)	0.63 (1.5%)	-	-	0.70 (1.5%)
		best	0.82(3.1%)	0.64(89.2%)	-	-	0.70(1.5%)
	AL(mean)	opt.	0.82(3.1%)	0.62(1.5%)	-	-	0.69(1.5%)
		best	0.83(43.1%)	0.63(60%)	-	-	0.70(60%)

Table 3.9: Results of the semi-supervised benchmark [86]. Percentage of query pool used to achieve the F_1 score is given in brackets.

	Method		KE	MW	WM	DW	micro F_1
H2	No Fine-Tuning		0.73	-	0.62	0.70	0.67
	Weak Transfer		0.59	-	0.42	0.73	0.58
	AL(max)	opt.	0.74(13.3%)	-	0.62(6.7%)	0.71(13.3%)	0.67(6.7%)
		best	0.79 (73.3%)	-	0.62(6.7%)	0.74(33.3%)	0.67(6.7%)
	AL(mean)	opt.	0.80 (20%)	-	0.62(6.7%)	0.71 (6.7%)	0.67 (6.7%)
		best	0.80 (20%)	-	0.62(6.7%)	0.73(20%)	0.67(6.7%)
H4	No Fine-Tuning		0.54	0.53	-	-	0.53
	Weak Transfer		0.59	0.65	-	-	0.63
	AL(max)	opt.	0.61(1.7%)	0.64(1.7%)	-	-	0.63 (1.7%)
		best	0.61(1.7%)	0.72 (67.2%)	-	-	0.65(67.2%)
	AL(mean)	opt.	0.58(8.8%)	0.63(10.5%)	-	-	0.61 (10.5%)
		best	0.60(52.6%)	0.70(66.7%)	-	-	0.65(66.7%)
H5	No Fine-Tuning		0.78	-	0.24	0.28	0.51
	Weak Transfer		0.79	-	0.32	0.28	0.55
	AL(max)	opt.	0.80(2.2%)	-	0.30(6.5%)	0.27(10.7%)	0.56(10.7%)
		best	0.80(2.2%)	-	0.36(95.6%)	0.28(50%)	0.57(54.3%)
	AL(mean)	opt.	0.80(2.2%)	-	0.34(26.1%)	0.28(4.3%)	0.56 (6.5%)
		best	0.80(2.2%)	-	0.34(26.1%)	0.29(52.2%)	0.56(6.5%)
H19	No Fine-Tuning		0.66	0.68	-	-	0.67
	Weak Transfer		0.75	0.69	-	-	0.71
	AL(max)		0.80(3.1%)	0.70(1.5%)	-	-	0.73(1.5%)
			0.81(64.6%)	0.71(29.2%)	-	-	0.73(1.5%)
	AL(mean)		0.78 (2.7%)	0.70(8.1%)	-	-	0.73(2.7%)
			0.79 (13.5%)	0.71(27%)	-	-	0.74(13.5%)

Table 3.10: Results of experimental Scenario 1. Percentage of query pool used to achieve the F_1 score is given in brackets.

The results of Scenario 2, with both strongly and weakly labelled data available during pre-training, and only weakly labelled data available during AL, are presented in Table 3.11. As expected, performance is improved over the baseline for all appliances in all of the houses. This is due to the presence of strong labels during the pre-training phase, ensuring that the model acquired more knowledge of appliance signatures, and consequently achieving better results with less weakly labelled samples added int he AL phase.

	Method		KE	MW	WM	DW	micro F_1
H2	No Fine-Tuning		0.78	-	0.78	0.84	0.82
	Weak Transfer		0.83	-	0.82	0.83	0.82
	AL(max)	opt.	0.82(6.7%)	-	0.80(6.7%)	0.83(6.7%)	0.82(6.7%)
		best	0.83 (13.3%)	-	0.82(46.7%)	0.84(93.3%)	0.82(6.7%)
	AL(mean)	opt.	0.83 (6.7%)	-	0.80(6.7%)	0.83 (6.7%)	0.82 (6.7%)
		best	0.84 (86.7%)	-	0.82(26.7%)	0.84(33.3%)	0.83(66.7%)
H4	No Fine-Tuning		0.71	0.69	-	-	0.69
	Weak Transfer		0.73	0.73	-	-	0.73
	AL(max)	opt.	0.76(6.9%)	0.84(5.2%)	-	-	0.81(5.2%)
		best	0.77(14%)	0.86(73.7%)	-	-	0.81(5.2%)
	AL(mean)	opt.	0.78(1.7%)	0.85(1.7%)	-	-	0.83(1.7%)
		best	0.78(1.7%)	0.86(28.1%)	-	-	0.83(1.7%)
H5	No Fine-Tuning		0.94	-	0.20	0.43	0.60
	Weak Transfer		0.95	-	0.41	0.55	0.70
	AL(max)	opt.	0.96(4.3%)	-	0.41(26.1%)	0.54(17.4%)	0.69(17.4%)
		best	0.96(4.3%)	-	0.42(76.1%)	0.57(60.9%)	0.72(65.2%)
	AL(mean)	opt.	0.96(2.2%)	-	0.36(28.3%)	0.51(2.2%)	0.67 (2.2%)
		best	0.96(2.1%)	-	0.40(39.1%)	0.58(28.3%)	0.71(28.3%)
H19	No Fine-Tuning		0.88	0.75	-	-	0.80
	Weak Transfer		0.76	0.69	-	-	0.71
	AL(max)	opt.	0.91(7.7%)	0.73(1.5%)	-	-	0.78(1.5%)
		best	0.94(72.3%)	0.73(29.2%)	-	-	0.78(1.5%)
	AL(mean)	opt.	0.89 (4.6%)	0.76(7.7%)	-	-	0.80(1.5%)
		best	0.89 (4.6%)	0.76(7.7%)	-	-	0.81(7.7%)

Table 3.11: Results of experimental Scenario 2. Percentage of query pool used to achieve the F_1 score is given in brackets.

Observed levels of uncertainty during the AL process are discussed next. Observed uncertainty levels of the whole query pool in House 4 are presented in histograms in Figure 3.10. In Scenario 1, only weak labels are available in the pre-training phase, and the model tends to be either very certain or very uncertain in its predictions. On the other hand, in Scenario 2, strong

labels are shown to the model during the pre-training phase, and the observed uncertainty levels are not as concentrated as in Scenario 1 - histogram for Scenario 2 is more flat, indicating more levels of uncertainty present. This is due to the fact that the model has seen multiple overlapping appliance activations in strong labels in the pre-training phase.

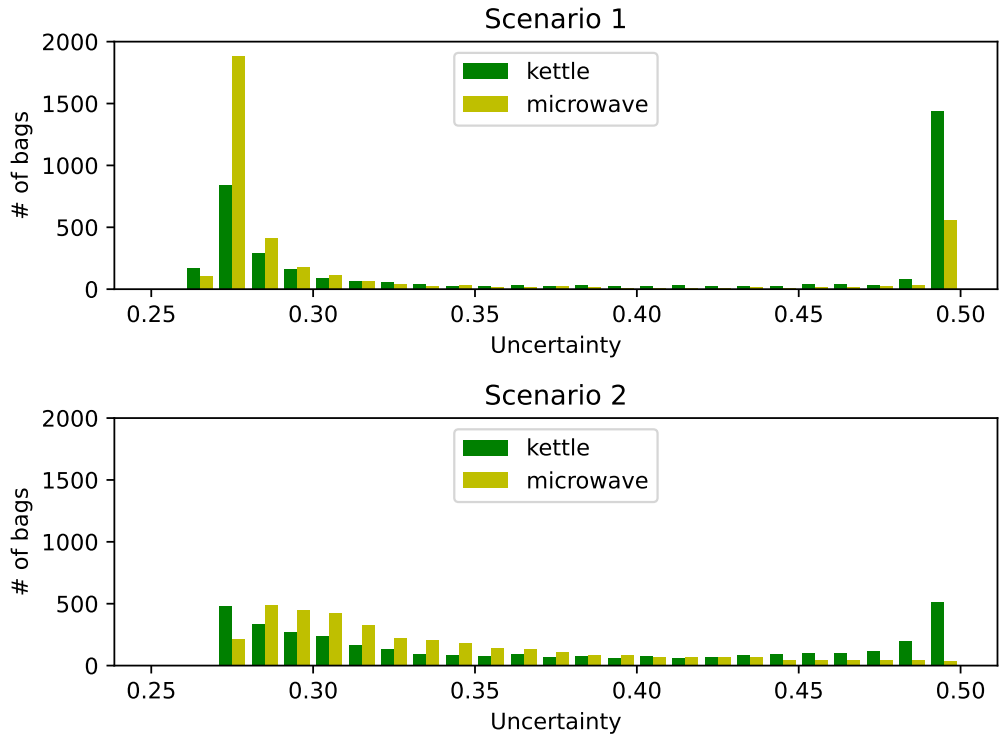


Figure 3.10: Uncertainty levels observed for the whole query pool for House 4 in Scenario 1 (top) and Scenario 2 (bottom).

Ratio of uncertainty levels for kettle and microwave appliances signal windows queried from House 4 at the beginning of AL process is shown in Figure 3.11: Scenario 1 with mean uncertainty across appliances used in acquisition function - top left; Scenario 1 with maximum uncertainty across appliances used in acquisition function - top right; Scenario 2 with mean uncertainty across appliances used in acquisition function - bottom left; Scenario 2 with maximum uncertainty across appliances used in acquisition function - bottom right. The uncertainty levels associated with microwave (light green)

are stacked to the uncertainty levels associated with kettle (dark green). Uncertainty is shown on the y-axis, while the x-axis presents signal windows present in a batch of 64 samples. Since kettle has more high-uncertainty signal windows, according to Figure 3.10, if using maximum uncertainty as an overall uncertainty measure in the acquisition function, signal windows are queried so that they have high uncertainty for kettle, but not necessarily for microwave. In opposition, if mean uncertainty is used as an overall uncertainty measure in acquisition function, then signal windows are queried so that they have high uncertainty for both appliances - kettle and microwave. Thus, mean uncertainty is a more reliable uncertainty measure, as described in Section 3.2.1.

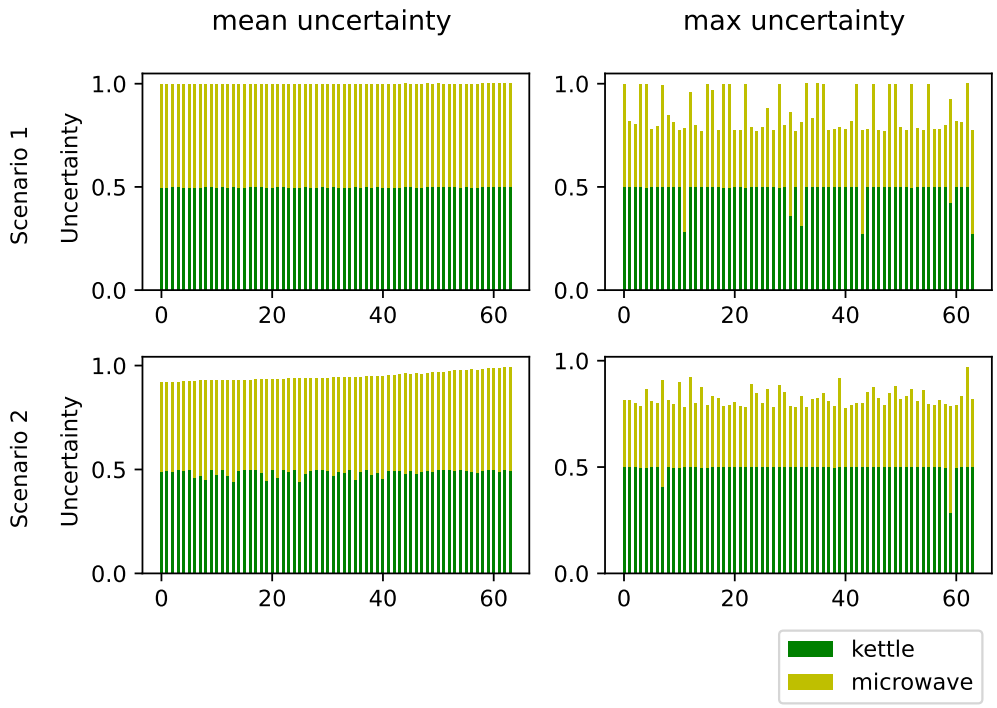


Figure 3.11: Observed ratio of uncertainty between kettle and microwave from House 4. Top row - Scenario 1; bottom row - Scenario 2; left column - mean uncertainty used in acquisition function; right column - maximum uncertainty used in acquisition function.

From the presented results, adding less data is sometimes better than adding more - with significantly reduced labelling effort at optimal points, performance is close to the best F_1 score. So, AL approaches are very useful in identifying high uncertainty data to include in fine-tuning and efficiently reduce labelling effort. Most importantly, weak labels can be successfully used to fine-tune pre-trained models with AL, and they can be easily obtained from end users based on the time when they used particular appliance, without the need for specific knowledge of appaliance signatures.

3.2.3 Summary

In this study a weakly supervised AL framework is proposed to successfully adapt pre-trained NILM models to new environments. Weak supervision offers the possibility of collecting labels from end users, through a dairy of appliance running times, as no sample-by-sample annotations are needed. AL ensures that as few queries as possible are made, which further reduces the labelling effort. We prove the efficiency of the proposed method under multiple experimental scenarios, with multiple appliances, across 4 test houses. Benchmark performance is exceeded with labelling effort reduced by 82.6-98.5%.

3.3 An active learning framework for microseismic event detection

Microseismic monitoring has been gaining attention over the past few years to further illuminate regional-scale induced earthquakes, termed microseismic events, to enable monitoring of subsurface projects, such as oil and gas production, hydraulic fracturing for unconventional resources, e.g., geothermal energy, or the reaction of the Earth's crust to impoundment and storage of water in dams. The injection of fluids into the ground during geothermal energy exploitation fractures the surrounding rock thereby inducing small earthquakes. These microseismic signals are characterised by very low signal to noise ratio (SNR) and hence, unlike earthquakes with relatively higher magnitude and SNR, these signals are challenging to detect in the presence of ambient noise from fluid injection and machinery. With computing resources becoming more and more available, deep neural network-based approaches on data from borehole arrays over the area of underground operations have

gained importance in induced microseismic event monitoring, as detailed in recent review paper [32]. These methods are used to provide better insights into underground processes for optimisation of the hydraulic fracturing during injection, as well as for real-time risk evaluation of induced seismicity. Microseismic monitoring starts with detection of microseismic events, which in turn enables localisation of hypocentres, and further characterisation of source mechanisms of microseismic events. The main issue of powerful deep-learning based detection methods, e.g., [33], [67] is that they require a large number of labelled samples, which is time-consuming and requires specialised knowledge. To alleviate this issue, in this study, we propose an AL strategy that works in conjunction with a deep learning-based algorithm, to include only most informative samples in the training set and thus reduce the labelling time. This in turn results in improving the speed and consistency of the base detection algorithm, which is a key requirement for microseismic applications [32]. Specifically, we adapt an AL framework described earlier in this chapter (Section 3.1.1), originally proposed for energy dissagregation, to microseismic event detection which creating high-quality training data sets in a data- and time-efficient way, by labelling only most informative samples. Furthermore, we transfer pre-trained models to new locations and different sensor types (where labelled data unavailable for pre-training), exploiting reliable predictions from multiple sensors at a time to make a final decision. Results show that AL brings improvement to the detection algorithm performing on both a new location and different sensor type, while saving 83% of labelling effort.

3.3.1 Methodology

The AL framework scheme used in this study is shown in Figure 3.12. The process works as described earlier, in Section 2.3. The acquisition function used is based on optimal thresholding, with stopping criterion as proposed in [10]. Namely, the deep learning model returns a prediction window containing values between 0 and 1, indicating if the event is detected at each timestamp inside the window. Then, the maximum of all prediction values in the window determines a single prediction value for the window. To determine if an event is detected or not, we set a threshold $0 < T < 1$. The interval $(0, 1)$ is split into three regions: a region $[0, T/2)$ where the model is certain that the window does not contain events, i.e., a negative prediction; a region where the model is uncertain about its prediction $[T/2, (1 + T)/2)$; and a region

$[(1 + T/2), 1]$ where the model is certain that the window contains an event, i.e., positive prediction. Then, samples belonging to the uncertain prediction region are ranked based on their distance from the decision threshold T, and a number $(batchsize/2)$ of all these uncertain samples that are the closest to T is queried. Additionally, a number of samples $(batchsize/4)$ is queried randomly from predictions that fall into certainly negative and a number $(batchsize/4)$ from certainly positive prediction regions. This constitutes a batch of samples that is queried, labelled and included in fine-tuning dataset. The largest number of samples is chosen from the uncertain region because those samples are supposed to bring the most information to the model. However, to prevent the model from overfitting and forgetting patterns of positive and negative samples for which it is typically certain about, samples are chosen from certain predictions as well. AL stops when there are three consecutive epochs with empty uncertain region.

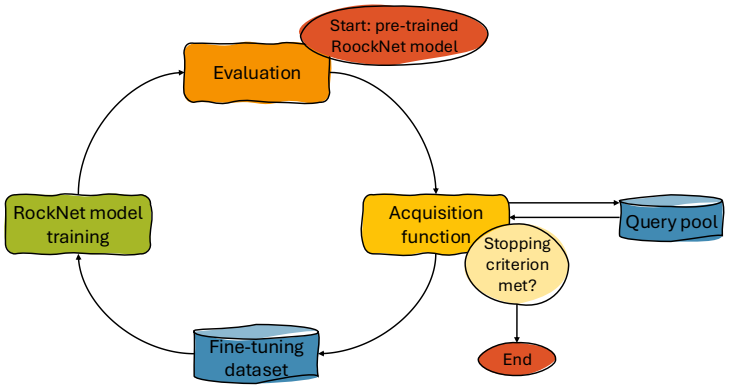


Figure 3.12: Active learning framework

To demonstrate efficiency of the proposed AL process, we use a popular RockNet network [67]. RockNet is a fusion model, taking both 3-channel time series window, and a spectrogram of the vertical channel of the window as inputs, and making a fusion of features extracted from both inputs. The model returns masks showing detected events - earthquakes and rockfalls. For earthquakes, masks are positive between P and S wave arrivals, and negative elsewhere. The model demonstrates excellent performance for detecting

earthquake and rockfall events [67]. In this study, we apply it to detect induced microseismic events, of much shorter duration and lower amplitudes, and also higher noise levels.

The dataset used in this study is recorded as part of the Utah Frontier Observatory for Research in Geothermal Energy (FORGE) project, as desribed in Section 2.7.2.

Experimental setup

Available recordings are divided into two periods - the last 15 minutes is reserved for testing, while the first 45 minutes are used as training set if pre-training, or query pool if performing AL. This 45-15 min split is the same for all sensors. Since for induced microseismic events time between P and S phase arrivals is very short (less than 100 ms), the input window length that is used in this study is 400 ms, or $W = 1600$ samples. Spectrograms are generated from vertical seismogram component of the input window, with the FFT length $N = 128$, and a step of $H = 10$ samples between STFT segments. So, the input dimensions are (3, 1600) for time-series input and (2, $N/2+1 = 65$, $W/H+1 = 161$) for the spectral domain input - one channel for real and one for imaginary spectrogram component. Since there is only one type of event in the dataset, at the output we only have one binary mask, of dimensions (1, 1600), which is positive between P and S wave arrivals, and negative elsewhere. Training windows overlap by 1/3, while testing windows do not overlap. For each sensor, there is 24,779 samples used as training set if the sensor is used for pre-training, or as query pool if AL has been applied to the sensor; and 2253 test samples. In the pre-training phase, training data was balanced by keeping an equal number of windows that do and do not contain an event. Excessive windows with no events present are discarded. Testing data was not balanced to reflect real-world recordings. Query pool data was also not balanced.

Three experiments were conducted in this research, to demonstrate value of proposed AL framework across different levels of generalisation to exploit multiple sensors in the borehole arrays on the site: (1) The deep learning model is pre-trained with data from two sensors from the well 78B-32 (sensors G7 and G8), and AL is performed with another sensor data from the same well (G5), to verify the AL approach. (2) The deep learning model is pre-trained with data from four sensors from the well 78B-32 (G5-8), and AL is performed with the data from a sensor from the well 58-32 (G5), to test

generalisation across different wells with the same instrument types - the two wells are distributed in different azimuths which can impact the signals. (3) The deep learning model is pre-trained with the data from 8 sensors from the wells 78B-32 (G5-8) and 58-32 (G5-8), and AL is performed with the data from a sensor from the well 56 (G2). This is to test generalisation across different wells and different instrument types, since well 56-32 has different instrumentation than the other two wells.

Model performance is evaluated using F_1-score as per Equation 2.4 from Section 2.8.

The stopping criterion as defined in [10], is based on the uncertain region of the acquisition function - if the region is empty for three consecutive AL iterations, it means that the model reached high levels of certainty and the AL process can be stopped.

Batch size used is 128, learning rate $1e-3$, and decision threshold $T = 0.5$ for Experiments 1 and 3, and $T = 0.6$ for Experiment 2. These model thresholds are heuristically set based on the testing data during the pre-training phase and are not further tuned during the AL phase. In the pre-training phase, training is performed for a maximum of 50 epochs with early stopping patience of 5. In the AL phase, each training is performed for 15 epochs, and the best model is used to make queries.

3.3.2 Results & Discussion

The results are presented in Figure 3.13 for the three experiments discussed in Subsection 3.3.1. AL iterations are presented on the horizontal axis, while the vertical axis shows the resulting F_1-score. Optimal points are marked by red dots.

In Experiment 1, where AL is used in the same well as the pre-training data comes from, AL curve rises the slowest - the data comes from the same source and is of similar quality as pre-training data, and there was enough pre-training data for the model to learn well. However, the performance does improve compared to the baseline model, and with significantly less data than if the whole query pool was labelled - only 15.5% of query pool is labelled (3840 out of 24779 samples, labelled over 30 iterations), saving 84.5% of labelling time. This indicates that microseismic event detection can benefit from AL to reduce labelling effort and improve performance.

In Experiment 2, which transfers a pre-trained model to a different well with the same type of sensors, AL gives promising results - $F1$-score is im-

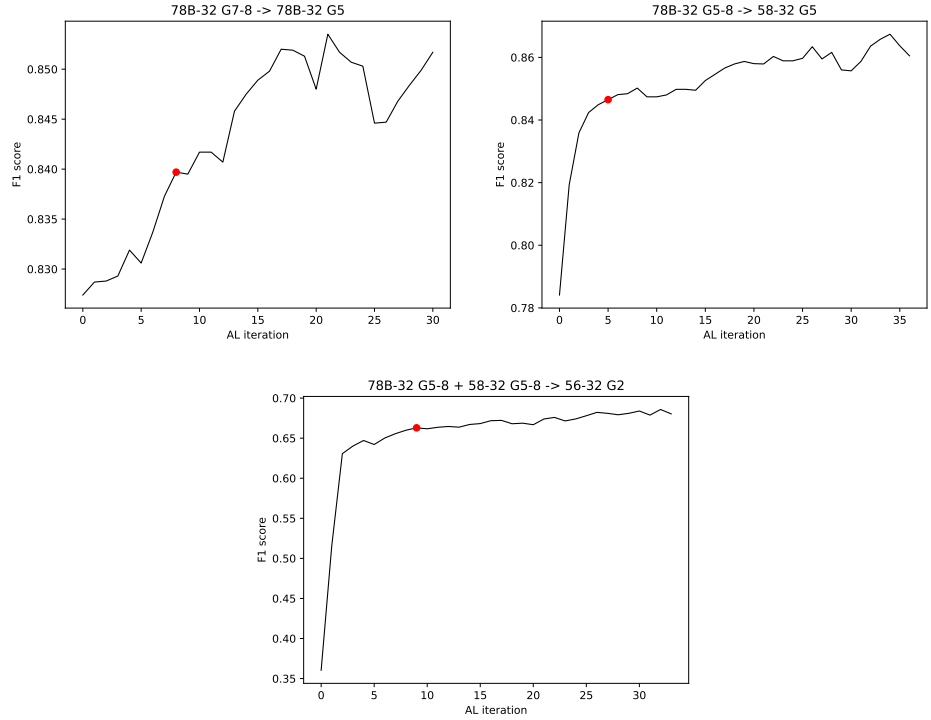


Figure 3.13: Results: Experiment 1 - top left; Experiment 2 - top right; Experiment 3 - bottom.

proved from 0.78 before any fine-tuning to 0.86 at the and of fine-tuning with AL, with only 18.6% of query pool labelled, reducing labelling effort by 81.4%. This scenario benefits more from AL because the new data used for fine-tuning during the AL phase is more informative due to environmental differences in two wells.

In Experiment 3, which transfers a pre-trained model to a different well with different measuring equipment, performance is the most improved. It was poor at the beginning ($F_1 = 0.36$), but with AL it is quickly improved after only a couple of iterations, reaching $F_1 = 0.68$ with 17% of data samples labelled, reducing labelling effort by 83%. This is due to the fact that newly introduced data is very different from the data used for pre-training, and AL provides significantly new information.

An example of model performance on a data sample from sensor G8 from well 58-32 in Experiment 2 is shown in Figure 3.14. The model has 92696

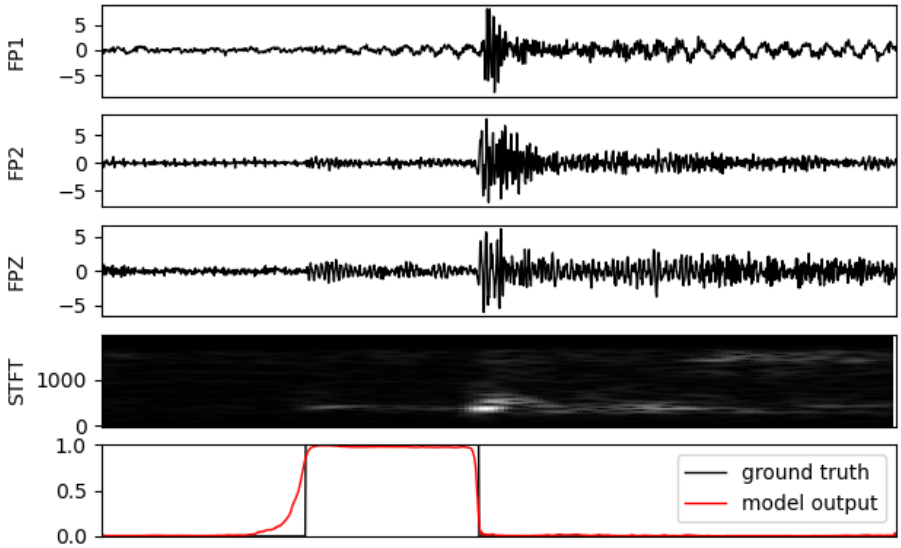


Figure 3.14: Model performance example - time-series input (FP1, FP2, FPZ), spectral input (STFT map), and ground truth and prediction for a data sample from sensor G8, well 58-32, Experiment 2.

trainable parameters, and processes on average 12.7 batches (12.7*128=1625.6 samples) per second in evaluation mode on a PC with 32GB RAM and an NVIDIA Titan Xp graphic card.

3.3.3 Summary

We demonstrated efficiency of AL for detection of induced micro-seismic events. We showed that by using the proposed approach to selecting most informative samples to be labelled, the labelling effort can be reduced by 80-84 % without affecting the performance. The proposed method is very efficient when transferred to a very different environment with different measuring equipment. It would be worth exploring AL with clustering approaches in future work, as well as usage of explanation tools which could inform the expert about the reasoning behind model decision and aid labelling.

Chapter 4

Human-in-the-loop active learning for time-series classification

As commonly done in the literature (and in the previous chapter), oracles are assumed within AL frameworks to provide absolutely true labels, without any errors, all with the same effort and at the same cost (see [14] for a review of AL approaches for medical image analysis - most of AL methods assume an oracle). This is a very unrealistic assumption - human error during labelling will be (unintentionally) introduced, especially for challenging to label samples (e.g., noisy samples) and time-series samples that are not always visually interpretable. Only a few studies have reported AL system results where users/experts are recruited to provide labels during the AL process. For example, [88] includes people in the labelling process, for an income prediction task using linear regression, investigating if AL could boost their trust and confidence in AI, depending on their level of familiarity with AI, and their willingness to engage with the process. However, studies that actually deploy AL concept focus mainly on social aspects of AL and human-computer interaction, e.g., trust, while using toy AL algorithms.

Human-in-the-loop approaches have not been explored in energy management related applications despite the acknowledged role of consumers on energy end use in order to meet European Green Deal Ambition goals related to bringing greenhouse gas emissions to the levels of 1990 by 2030 [89].

Building on the prior work described in Section 3.1.1 that proposes a framework for AL for low-frequency model-based NILM, assuming perfect

error-free labelling, in this study we propose a novel human-in-the-loop approach which sits between active and interactive learning, where the machine selects examples to query, then through a user interface which shows the time-series electrical signal under questions, a human expert manually labels such examples. Due to the nature of the variable electrical signals belonging to the same class, we show that human uncertainty is possible and the model learns incrementally until a stopping criterion is met. Our approach is demonstrated for the problem of energy disaggregation from widely available smart meter aggregate measurements, i.e. NILM, which suffers from unavailability of labelled samples (i.e., labelled appliances contributing to the aggregate at each sampling point).

4.1 Methodology

In this section we describe the proposed AL approach, illustrated in Figure 4.1. As in Chapter 3, Algorithm 1, described in Section 2.3, is used to select samples to query. Four main contributions to work reported in Chapter 3 are made. First, a new acquisition function $q(\cdot)$ is proposed based on hypothesis testing to ensure diversity of labels in terms of reliability and classes (see Subsection 4.1.1). Second, a stopping criterion is introduced when all "uncertain" samples are exhausted (see Subsection 4.1.2). Third, confidence levels are included during model learning within the fine-tuning step (Subsection 4.1.3), to account for experts' confidence about provided labels and mitigate the effect of errors introduced for hard-to-label samples. Finally, after the fine-tuning step, an additional step for returning potentially wrongly labelled data samples back to experts for re-labelling is proposed (Subsection 4.1.4).

4.1.1 Acquisition function

Traditional uncertainty-based acquisition strategies for selecting samples to label tend to first query windows of samples containing appliance activations, i.e., positive samples [10]. This leads to a very unbalanced set after labelling, containing predominantly positive samples. To keep the diversity of queried samples, both in terms of classes (all classes should be well represented) and model uncertainty (most uncertain samples should be queried), a new acquisition function based on maximum a posteriori (MAP) hypothesis testing is

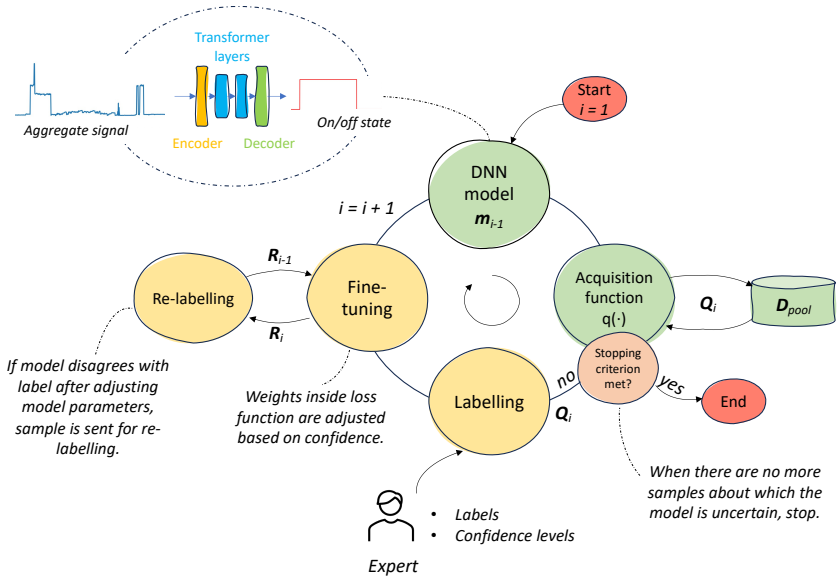


Figure 4.1: Active learning framework.

proposed next.

Let $\hat{y}$ be a realisation of a random variable $\hat{Y} \in [0, 1]$ denoting the model output (0=appliance if off; 1=appliance is on). Let us consider two hypotheses: hypothesis H_0 corresponding to the appliance being in off-state, and hypothesis H_1 corresponding to the appliance being in on-state. Suppose that prior probabilities of both states are known, i.e., $P(H_0)$ and $P(H_1)$, as well as probability density distributions of model output $\hat{y}$ under the two hypotheses, i.e., $f_{\hat{Y}}(\hat{y}|H_0)$ and $f_{\hat{Y}}(\hat{y}|H_1)$.

Then, after applying Bayes' rule, posterior probabilities of hypotheses H_0 and H_1 are obtained as:

$$P(H_i|\hat{Y} = \hat{y}) = \frac{f_{\hat{Y}}(\hat{y}|H_i) \cdot P(H_i)}{f_{\hat{Y}}(\hat{y})}, i \in \{0, 1\}. \tag{4.1}$$

Using the MAP test, the winning hypothesis will be the one that maximises (4.1). Since the denominator is the same for both hypotheses, hypothesis H_0 is chosen if and only if:

$$f_{\hat{Y}}(\hat{y}|H_0) \cdot P(H_0) > f_{\hat{Y}}(\hat{y}|H_1) \cdot P(H_1). \tag{4.2}$$

Otherwise, hypothesis H_1 is chosen.

The model output value $\hat{y}^*$ for which posterior probabilities of the two hypotheses, H_0 and H_1, are the same, i.e.,

$$f_{\hat{Y}}(\hat{y}^*|H_0) \cdot P(H_0) = f_{\hat{Y}}(\hat{y}^*|H_1) \cdot P(H_1) \tag{4.3}$$

is considered the most challenging model output value to make a decision. Therefore, model output space $[0, 1]$ is divided into three regions: likely negative model predictions (H_0 chosen; model output value close to 0), likely positive model predictions (H_1 chosen; model output value close to 1), and uncertain model predictions (model output value close to $\hat{y}^*$ where posterior probabilities for H_0 and H_1 are equal). See Figure 4.2 for illustration. Each point in the model output space is assigned to one of three regions depending on its proximity to 0, $\hat{y}^*$, and 1. Samples to be queried are taken from all three regions as per equation:

$$\begin{aligned}
\boldsymbol{Q}_i &= \boldsymbol{Q}_{i,\ likely\ negative} \cup \boldsymbol{Q}_{i,\ uncertain} \cup \boldsymbol{Q}_{i,\ likely\ positive} \\
\boldsymbol{Q}_{i,\ likely\ negative} &\subset \{s \in \boldsymbol{D}_{pool} | y = m_{i-1}(s) \in (0, \frac{\hat{y}^*}{2})\} \\
\boldsymbol{Q}_{i,\ uncertain} &\subset \{s \in \boldsymbol{D}_{pool} | y = m_{i-1}(s) \in (\frac{\hat{y}^*}{2}, \frac{1+\hat{y}^*}{2})\} \\
\boldsymbol{Q}_{i,\ likely\ positive} &\subset \{s \in \boldsymbol{D}_{pool} | y = m_{i-1}(s) \in (\frac{1+\hat{y}^*}{2}, 1), \}
\end{aligned} \tag{4.4}$$

where query from the current iteration i is denoted by $\boldsymbol{Q}_i$. $\boldsymbol{D}_{pool}$ is query pool, s denotes samples belonging to the query pool, m_{i-1} is the model from previous AL iteration, and y is the model output for sample s. The number of samples from each region is controlled by hyper-parameters.

Since off-state of an appliance is more frequent than on-state (that is, most appliance are not used continuously), point $\hat{y}^*$ is expected to be closer to 1 than to 0 (see the example in Figure 4.2), so samples containing measurements while appliance is turned on are favoured by this strategy, which is beneficial to NILM algorithms, as discussed later in Section 4.3.1. Most of queried samples therefore come from the uncertain region as defined above (the number is controlled by a hyper-parameter), but to prevent model from forgetting, samples are also taken from the two likely (positive/negative) regions.

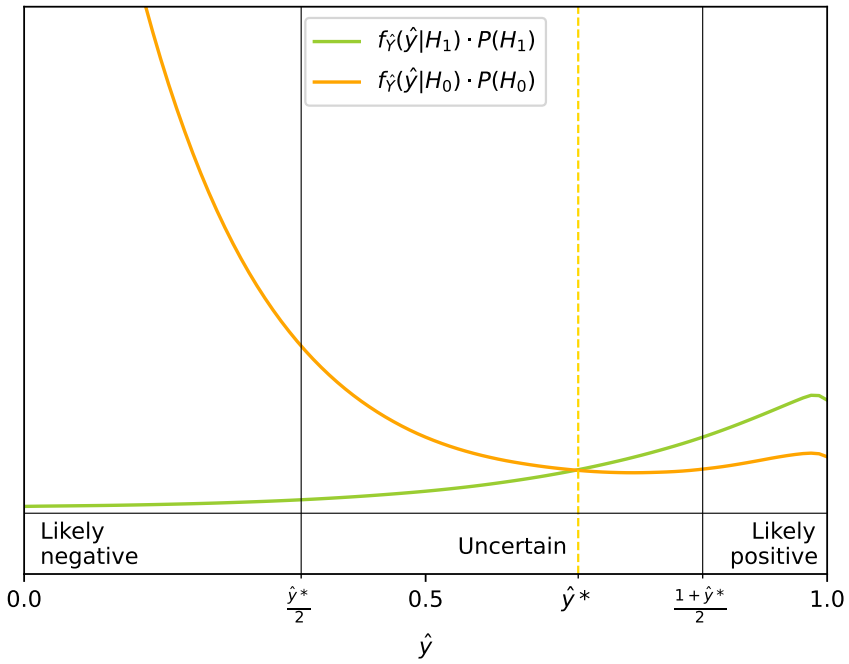


Figure 4.2: Acquisition strategy - an illustration (for appliance kettle): Distributions of the model output under hypotheses H_0 and H_1, and three model output space regions.

4.1.2 Stopping criterion

Stopping criteria, as discussed in Chapter 2, Section 2.3, usually rely on comparison of performance across subsequent AL iterations (e.g., in [18]) or on agreement of models in subsequent iterations (e.g., in [20] and [19]). To avoid the need to store the models from multiple iterations and compare them, or to store the model outputs or uncertainty levels from multiple iterations, which can be resource-intensive, a stopping criterion relying on confidence of the model from a single, current iteration is designed.

When using the proposed acquisition function, as described in Subsection 4.1.1, there is a region in the model output space where model predictions are considered uncertain. During the AL process, the uncertain region is quickly exhausted, but, as the model changes during the process, the model

output for some samples can shift from likely positive or negative regions to uncertain. When samples from the uncertain region are exhausted, the process is meant to stop - it means that the uncertain samples have been already included in training and only samples for which the model has high level of certainty remain. To ensure that the model is consistently certain in its predictions, patience for a few epochs can be introduced - i.e., AL can stop when the uncertain region is empty, or does not contain enough samples to fill $\boldsymbol{Q}$ for a few consecutive epochs, as per Equation 4.5.

$$patience_i = \begin{cases} patience_{i-1} + 1, & \{s \in \boldsymbol{D}_{pool} | y = m_{i-1}(s) \in (\frac{\hat{y}^*}{2}, \frac{1+\hat{y}^*}{2})\} = \emptyset \\ 0, & \{s \in \boldsymbol{D}_{pool} | y = m_{i-1}(s) \in (\frac{\hat{y}^*}{2}, \frac{1+\hat{y}^*}{2})\} \neq \emptyset \end{cases}$$
$$S = \begin{cases} False, & patience_i < max\_patience \\ True, & patience_i \geq max\_patience \end{cases} \tag{4.5}$$

Patience in current iteration i is denoted by $patience_i$, while $patience_{i-1}$ denotes the patience from the previous AL iteration. S is a boolean variable denoting if the stopping criterion has been met or not. This strategy offers timely stopping of the AL process without the need to store and compare performance of the models from earlier stages of the process. In addition, this strategy eliminates the need for setting a predefined threshold on model performance, which can be a challenging task since it is not always straightforward to estimate the level of expected performance if the model is deployed in a new previously unseen environment.

4.1.3 Exploiting experts' confidence

To account for possible wrong labels introduced by humans during labelling, a method to incorporate their confidence about a label is introduced. Expert confidence levels are used to set weights inside the loss function during training - instead of treating all samples equally - by applying weighted average when calculating the loss as:

$$Loss = \frac{1}{N} \cdot \sum_{i=1}^{N} c_i \cdot Loss_i. \tag{4.6}$$

Here, N denotes the total number of samples, and $Loss_i$ is the model's loss value for the i-th sample. The higher the expert certainty, the higher

the sample confidence weight c_i. A lower weight means that the effect of a sample to the calculated loss is attenuated, thus it contributes less to model learning. [1]

4.1.4 Re-labelling samples

To reduce likelihood of training the model with wrong labels, a mechanism for returning samples with possibly erroneous labels, $\boldsymbol{R}$, for re-labelling is implemented as:

$$\boldsymbol{R} = \{s \in \boldsymbol{Q} : MR(y, \hat{y}) < T_{\text{return}}\} \tag{4.7}$$

where

$$MR(y, \hat{y}) = \frac{\sum_{i=1}^{N} \min\{y_i, \hat{y}_i\}}{\sum_{i=1}^{N} \max\{y_i, \hat{y}_i\}}, \tag{4.8}$$

and N is the signal window length.

Namely, after the loss function has been applied to each newly added sample $s_i \in \boldsymbol{Q}$, match rate (MR, Equation 4.8) between the correct label y_i of sample s_i and soft model prediction $\hat{y}_i$ is calculated - if MR is below a threshold T_{return} even after the loss function is applied, it means that the sample possibly deviates from the rest of the training set, and that the label is possibly wrong; thus this sample is sent back for re-labelling, enabling the expert to re-consider and change their original decision.

4.2 Experimental Setup

4.2.1 Data & DNN model

To facilitate reproducibility, we use the well documented public REFIT [52] and UK-DALE [53] real-world electrical load measurements datasets, as described in Section 2.7.1.

In all experiments, as in [70], REFIT house 5, and UK-DALE house 1, which contain all four targeted appliances with many activations, are used for testing. A continuous period without missing data from 1st March 2014 to 1st September 2014 is chosen - first 2 months for the query pool and the rest

[1] The exact way of setting labels and confidence levels through a user interface for the application considered in this study is described in Section 4.2.3.

Appliance	Training houses (REFIT)
Kettle	6 (28.11.2013-28.06.2015.)
	8 (01.11.2013-10.05.2015.)
	17 (06.03.2014-19.06.2015.)
Microwave	6 (28.11.2013-28.06.2015.)
	8 (01.11.2013-10.05.2015.)
	17 (06.03.2014-19.06.2015.)
Washing machine	2 (17.09.2013-28.05.2015.)
	3 (25.09.2013-02.06.2015.)
	16 (10.01.2014-08.07.2015.)
Dishwasher	2 (17.09.2013-28.05.2015.)
	3 (25.09.2013-02.06.2015.)
	16 (10.01.2014-08.07.2015)

Table 4.1: REFIT houses and time periods used for training for each target appliance.

for testing, to ensure that there is enough diversity among testing data, and that the query pool is of reasonable size since manual labelling is included in experiments. Continuous recordings from the query pool and testing data are sliced into non-overlapping windows before being fed to the model. As explained in Subsection 2.3, labels are not available for the query pool data, so, in the query pool, only aggregate electricity consumption measurements are used. Labels are provided later after the model makes a query, either by an oracle (Experiment 1), or by an expert (Experiment 2). For testing, submetering measurement labels are used to quantify model performance. Houses and time periods used for pre-training of each appliance are shown in Table 4.1 - for washing machine and dishwasher as in [70], and for microwave and kettle as in [44]. It is worth mentioning that in NILM, like in many other real-world applications based on time-series data where class-balance depends on the frequency of events, even though raw measurements are highly imbalanced (home appliances are turned off most of the time), it is possible to create balanced training datasets through continuous recording over long periods of time, without data augmentation.

The DNN model used in this paper is the ELECTRIcity transformer [70], designed to work well with unbalanced data. The model architecture is presented in Figure 4.3. It is trained in two phases: an unsupervised pre-training

phase followed by a supervised training phase. The model shows superior performance to other state-of-the-art algorithms [70]. In experiments in this paper, for creating pre-trained models to be transferred to a new house, both training phases are used, but during the AL process, only supervised fine-tuning phase is used. A *sigmoid* activation function has been added to the final layer of the network to perform on/off-state binary classification (instead of regression as in [70]). One DNN model is created per monitored appliance - for example, if 4 different appliances are monitored in a house, then 4 different models will be created, for determining the state of each appliance separately. Therefore, each DNN model performs classification to 2 classes - on and off state. Since the model works in a sequence-to-sequence fashion, a pooling function is applied to the model output to get a single uncertainty value, by taking the maximum value of the model prediction window, with a reasoning that signal window is considered positive if there is at least one sample in that window where the appliance is active.

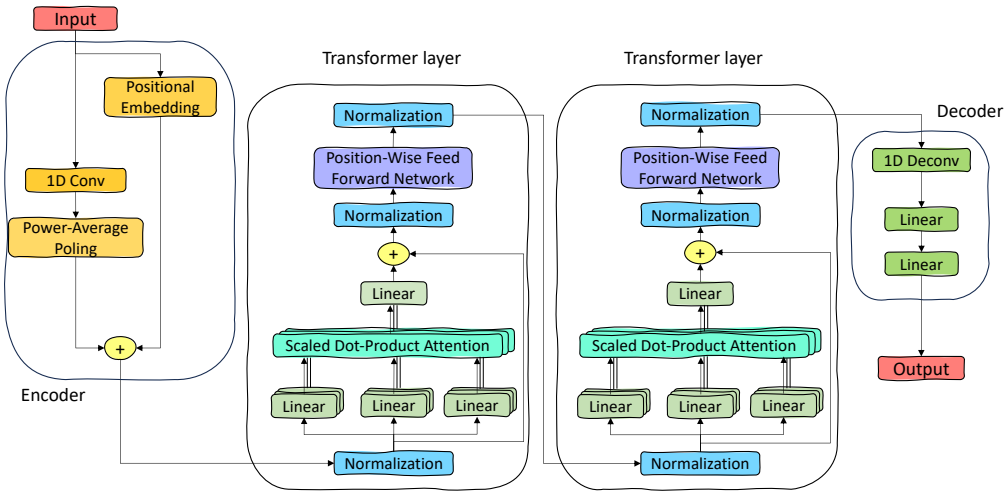


Figure 4.3: Architecture of ELECTRIcity transformer model [70].

The classification performance of the DNN-based NILM algorithm is evaluated using the standard F_1-score, as per Equation 2.4.

AL performance is usually presented as a curve showing model accuracy against the number of labelling iterations, i.e., the number of samples queried

and labelled. If a point with no labelling effort (i.e., iteration 0; 0 labelled sata samples), and the maximum possible model performance (i.e., F_1-score equal to 1) is considered as an "ideal" point, then the optimal point of the AL process can be calculated as the point with minimum Euclidean distance from the ideal point (see Eq. 3.3).

4.2.2 Experiments

Two experimental settings were considered in this study, as described next.

- Experiment 1: Transfer learning with labels obtained via submetering, with simulated labelling errors and re-labelling mechanism, and simulated confidence levels

  In this experiment, samples from the query pool are labelled using submetering electricity consumption measurements. The effect of balancing of queried batches using different acquisition functions is explored using several balanced acquisition functions. Stopping criterion is also applied to reduce labeling effort after the optimal point is achieved, as explained in Subsection 4.1.1. To study the effect of possible labelling errors and mimic a real-world AL process when labels are provided by humans, different levels of false positive and false negative errors are simulated. Namely, if the model prediction for a sample in the query pool contains appliance activation, but the ground truth does not, false positive error is introduced to that sample by accepting model prediction as ground truth label, with a predefined probability. On the other hand, if model prediction for a sample does not contain appliance activation, but the ground truth does, false negative error (missing appliance activation; setting ground truth label to 0) is introduced with a predefined probability. The proposed re-labelling mechanism (Section 4.1.4) is then applied to detect possibly wrong labels and send them back for re-labelling. Also, simulated confidence levels in correlation with simulated errors were utilized throughout the process to attenuate negative effects of errors (Section 4.1.3).

- Experiment 2: Transfer learning with expert labelling, exploiting expert confidence levels

  In this experiment, the best setup obtained from the first experiment is verified in a real-world scenario, where experts provide labels during the

AL process. As those labels can be erroneous, expert's confidence level is considered during the training phase, assuming that if an expert is not confident about a label, the label is more likely to be wrong, and should be used with caution. A graphical user interface enabling experts to quickly provide labels together with their confidence was developed and used (see Section 4.2.3).

All DNN and AL hyper-parameters are shown in the Table 4.2. Parameters for the DNN used are set as in [70]. Although in [70], a window length of 480 samples is used for all appliances, here the window length is shortened for kettle and microwave to 120 samples instead of 480, because those appliances have very short activation times. Therefore query pool sizes differ for kettle and microwave (4416 samples) from those for washing machine and dishwasher (1104 samples), although the same time period of two months is used for the query pool. Learning rate and the number of epochs are different in the pre-training and fine-tuning phases - they are set lower in the fine-tuning phase within the AL process to mitigate effects of overfitting due to a small number of labelled samples, especially in the beginning. At each labelling iteration, one batch of samples is queried. Confidence threshold for stream-based uncertainty acquisition function is set to be the same as in Section 3.1. The number of uncertain samples coming from the uncertain region for the proposed acquisition function is set to 56 so that the majority of queried samples come from the uncertain region, and the rest - 8 samples per iteration from the likely positive and likely negative prediction regions - for the purpose of preserving diversity among queried data and preventing forgetting of the model. A PC with the following specifications is used in the experiments: Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz, 32GB RAM, and a NVIDIA TITAN Xp GPU.

4.2.3 User interface

In order to facilitate experts' participation in the AL process, a graphical user interface, shown in Figure 4.4, is developed. [2] Queried samples (windows of electric load measurements) from one labelling iteration are shown to the expert in a sequence, one by one. The aggregate signal in Watts is shown

[2]The user interface described here is designed specifically for the problem of NILM considered in this paper. General idea and methodology of using expert-provided labels and confidence is given in Section 4.1.3.

DNN model	
input window size	kettle, microwave: 120 washing m., dishwasher: 480
heads, hidden, layers	2, 256, 2
dropout rate	0.2
tau	0.1
learning rate	pre-training: 1e-3 fine-tuning: 1e-4
epochs	pre-training: 100 fine-tuning: 10
batch size	64
model threshold	0.3

Active learning	
queries per iteration	64
query pool size	2 month worth of samples: kettle, microwave: 4416 washing m., dishwasher: 1104
confidence threshold (stream-based unc. acq. function; Exp.1)	0.9
# of samples for the proposed acquisition function	4 likely neg. 56 uncertain 4 likely pos.

Table 4.2: Hyper-parameters used in the experiments.

on the left vertical axis - this value can help experts decide if the appliance in question is on or off. Model prediction is shown together with aggregate signal (the values of the prediction can be seen from the right vertical axis, in range 0-1), to inform experts of model's behaviour and possibly help them make a decision. Horizontal axis shows time, which also can help an expert make a decision - e.g., some appliances are more likely to be operated during a particular time of a day. Experts are asked to mark the part of the window where they think the appliance of interest is active, by simply drawing a rectangle over that area, as shown in Figure 4.4. Apart from labels, experts are asked to provide their confidence level associated with each label - i.e., they are asked to select one of three offered options - low confidence, medium confidence or high confidence. High confidence is then mapped in the back end to a coefficient $k = 3$, mid confidence to $k = 2$, and low confidence to $k = 1$, which are then converted into sample weight according to:

$$c_i = \frac{N}{\sum_{j=1}^{N} k_j} \cdot k_i, \tag{4.9}$$

calculated at a batch level. This way, the samples with higher confidence have triple the weight of samples with lower confidence, and samples with mid confidence double, but the sum of weights in a batch remains the same as before the weights were adjusted. Obtained weights are then included in the loss function (see Eq. 4.6), as described in Section 4.1.3.

Nonetheless, experts using the interface may inadvertently introduce label noise—such as misclicks or ambiguous interpretations—while assigning labels or confidence scores to each data sample, potentially negatively affecting model's training and the active learning process.

4.3 Results & Discussion

In this section we report our experimental results. The goal of the experiments is to: (1) evaluate performance of the proposed acquisition function against state-of-the-art benchmarks without labelling errors; (2) test effectiveness of the proposed stopping criteria; (3) test if the proposed re-labelling leads to performance gains, and (4) show usefulness of the introduced expert confidence scores.

We organise the section into two parts: first we report the results related to Experiment 1 as described in the previous section; then, we evaluate the

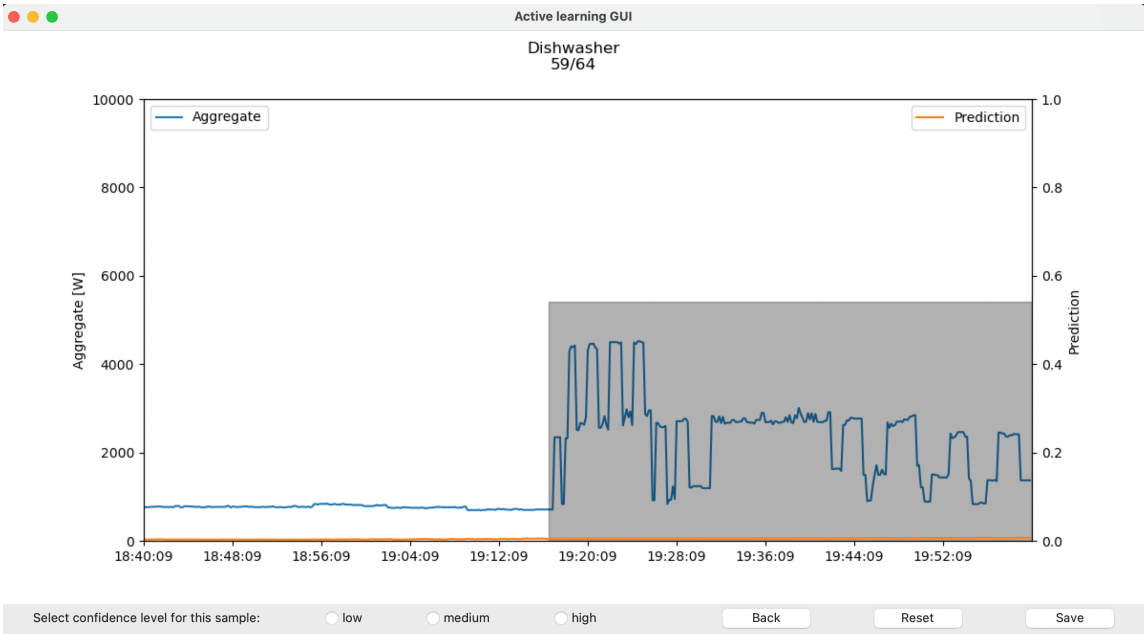


Figure 4.4: User interface that facilitates quick labelling by experts participating in the AL process.

proposed system with three NILM experts using the designed user interface.

4.3.1 Experiment 1

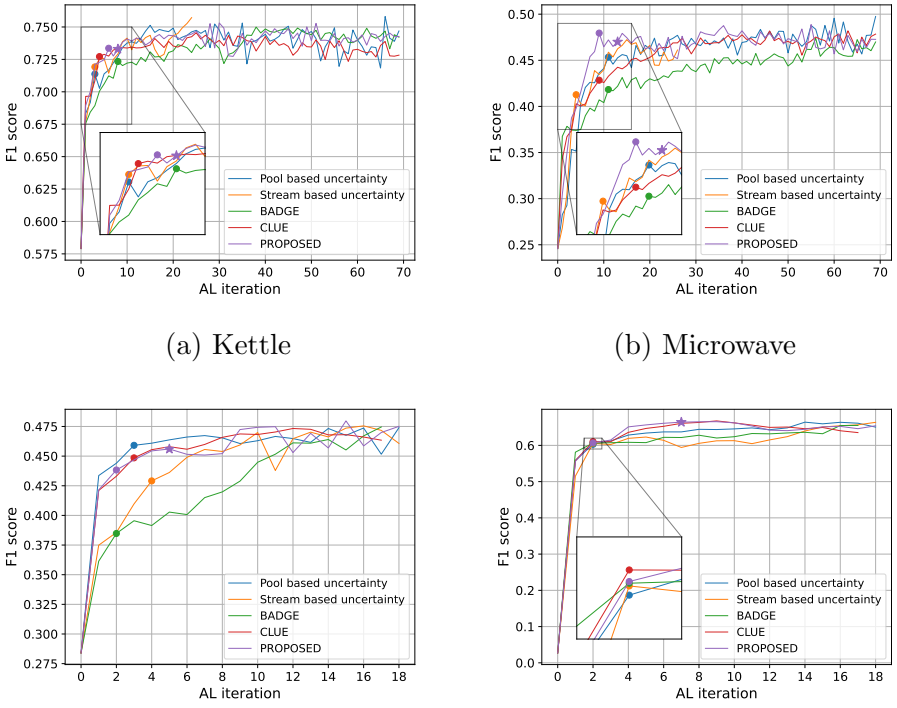
Acquisition function

In this subsection we compare the performance of the proposed acquisition function against state-of-the-art benchmarks. Acquisition functions used for benchmarking are pool- and stream-based uncertainty acquisition functions, as they are lightweight algorithms and demonstrate good performance for the NILM problem (see Section 3.1).

For the stream-based uncertainty acquisition function an informativeness threshold is used to make a decision if samples are sent for labelling or not (see Section 2.3). Since in Section 3.1, it was demonstrated that low values of informativeness threshold provide higher improvement in the beginning of the AL processes, the starting threshold is set to 0.9, and then as the process progresses, it is increased if the number of selected samples is lower than the batch size. This way the AL process experiences both high performance im-

provement in the beginning and longer lasting process which includes samples with a higher confidence at later stages.

Two additional benchmarks are used that attempt to diversify samples and balance the classes: BADGE acquisition function [11] that diversifies queried samples to avoid redundancy by looking at gradient embeddings, and CLUE acquisition function [12], that diversifies queried samples by looking at penultimate layer activations, but also includes least confidence uncertainty, i.e., it takes advantage of both uncertainty and diversification of queried batch of samples.



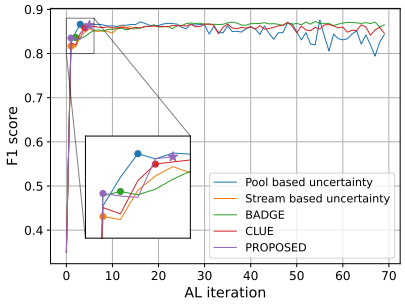
(a) Kettle

(b) Microwave

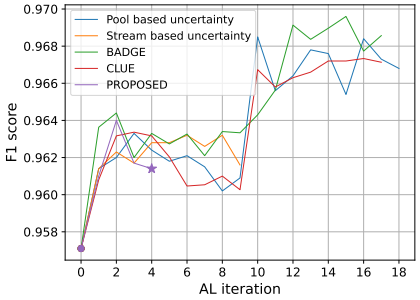
(c) Washing Machine

(d) Dishwasher

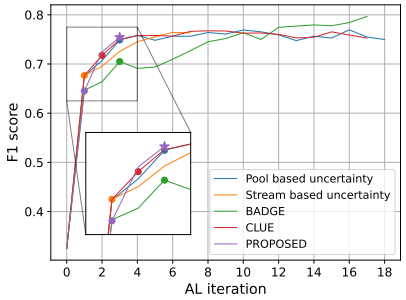
Figure 4.5: Comparison between different acquisition functions - transfer to REFIT house 5: the proposed one based on the optimal thresholding strategy; pool-based uncertainty (as in [10]); stream-based uncertainty [10]; BADGE [11]; CLUE [12]. Dots denote the optimal points and stars the stopping point for the proposed strategy.



(a) Kettle



(b) Washing Machine



(c) Dishwasher

Figure 4.6: Comparison between different acquisition functions - pre-training on the REFIT dataset and transfer to UK-DALE house 1: The proposed acquisition function based on the optimal thresholding strategy; pool-based uncertainty (as in [10]); stream-based uncertainty [10]; BADGE [11]; CLUE [12]. Dots denote the optimal points and stars the stopping point for the proposed strategy.

Results of the comparison for the four appliances from REFIT house 5 are shown in Figure 4.5, and from UK-DALE house 1 in Figure 4.6. Horizontal axis shows the AL, i.e., labelling iteration, and vertical axis the achieved F_1-score. Optimal points calculated based on Eq. 3.3 are marked as dots, and stopping points for the proposed acquisition function as proposed in Subsection 4.1.2 are marked with stars. Results for BADGE [11] and CLUE [12] acquisition functions are averaged over 3 independent runs, because those algorithms depend on cluster initialisation.

Numerical results of this experiment for REFIT house 5 - achieved F_1-

scores and percentage of query pool samples queried at optimal point, maximum performance point and stopping point (for the proposed acquisition function) - are presented in Table 4.3, and for UK-DALE house 1 in Table 4.4.

As shown in Figures 4.5 and 4.6 (and in accordance with the findings from Section 3.1), pool- and stream-based acquisition functions both demonstrate high and stable performance. Batch-aware acquisition function BADGE [11] performs slightly worse than pool- and stream-based uncertainty (except for dishwasher in Figure 4.5d, and washing machine in Figure 4.6b), which indicates that the dataset does not benefit from batch balancing during acquisition, and that some types of samples (windows containing activations in this case) are more significant for model improvement. Although CLUE [12] diversifies queried samples as well, it exploits model uncertainty, so its performance is on par with pool-based acquisition function.

It is observed that with pool- and stream-based uncertainty acquisition functions, in the beginning of the process, mostly samples containing appliance activations are being queried and added to the training set, due to high uncertainty associated with them. That usually results in a large jump in performance. After all samples containing activation have been exhausted, samples without activation, but with high aggregate consumption, are being queried, and finally, samples without appliance activation and with low aggregate values are being queried.

Our proposed acquisition function favours low- and mid-certainty signal windows containing appliance activation, but also chooses samples without appliance activation, as well as high-certainty samples containing activation. That way, it keeps diversity among queried data, but also ensures that sufficient number of samples important for learning of new patterns are regularly selected. This strategy performs the best for kettle and microwave in REFIT house 5 (Figures 4.5a and 4.5b), since these two appliances are often confused with washing machine, since they have similar wattage. Moreover, those two appliances have very short duration times, hence a small number of samples within a window contain an activation. Thus, it is important to choose enough samples where the model predicts there is an activation, but also high-certainty samples help in preventing forgetting of patterns of interest, and correcting wrong behaviour caused by confusions with other appliances as described above.

For washing machine and microwave in REFIT house 5, with multi-state relatively more complex signatures, high-certainty samples are usually cor-

rectly predicted and the model benefits mostly from low-certainty samples. Therefore, the pool-based acquisition function performs well. Transferability of the washing machine and microwave models, compared to more distinct kettle signature, is relatively poor in general [56, 80] and often excluded in the NILM literature.

However, for washing machine in UK-DALE house 1, initial performance is very good, due to much lower background noise levels in this house. The AL curve, hence, does not have the usual shape, but its range covers only the F_1-score from 0.96 to 0.97 - there is not much room for improvement if starting performance is so good, as opposed to other appliances from this house and from REFIT house 5.

For the dishwasher, the starting performance is poor in REFIT house 5 and UK-DALE house 1, indicating that the dishwasher model in the test houses is very different from those present in the pre-training dataset, but only two (Figure 4.5d) and three (Figure 4.6c) AL labelling iterations are sufficient to significantly improve the performance. All query strategies perform equally well in REFIT house 5 - due to a very low starting performance, all acquisition functions provide a highly informative fine-tuning set that contributes to significant model improvement. Nevertheless, it can be seen that the proposed strategy (purple star in Figures 4.5d and 4.6c) led to the highest performance in both test houses.

Based on the proposed stopping criterion, stopping is applied after 3 consecutive iterations with less than a half of the required high-uncertainty samples present in the query pool, to ensure consistent certainty of the model. Stopping points are therefore always located several iterations after the optimal points. It can be seen from Figures 4.5 and 4.6, as well as from the numerical results presented in Tables 4.3 and 4.4, that the proposed early stopping significantly saves the labelling effort with negligible performance loss. Indeed, the gap between the point where the maximum performance is achieved and the stopping point is always very small.

The impact of errors and re-labelling mechanism

Next, we evaluate the performance when labelling errors are present in REFIT house 5 and assess usefulness of the proposed re-labelling strategy with the proposed acquisition function and the proposed stopping criteria.

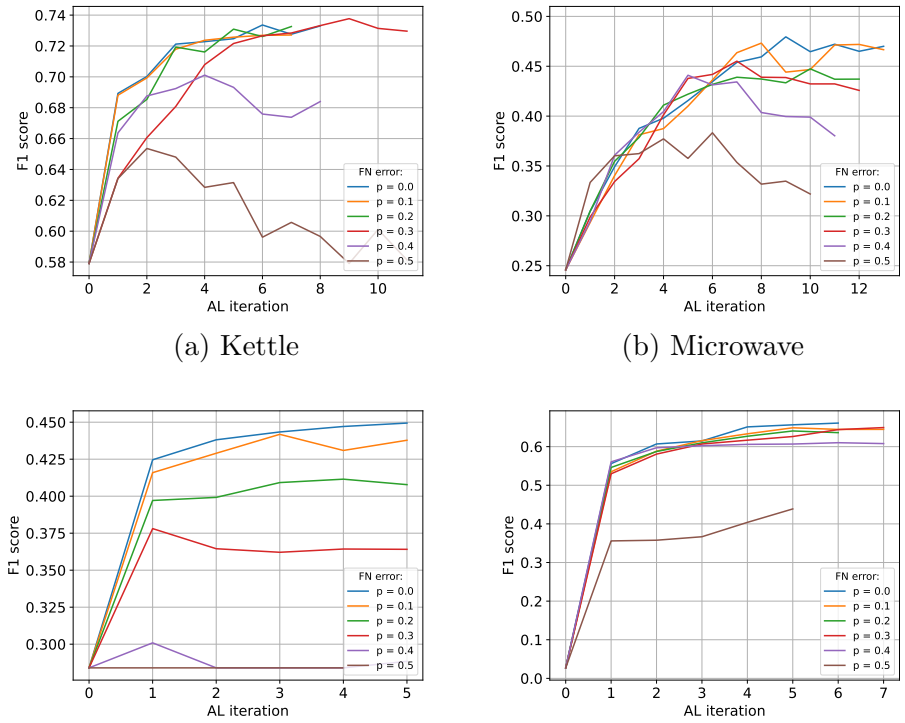
Figure 4.7 shows the results when false negative errors are introduced into labels, i.e., positive labels are set as negative. Blue line corresponds to

Acquisition function		Kettle		Microwave		Washing M.		Dishwasher	
		F_1	$\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$	F_1	$\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$	F_1	$\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$	F_1	$\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$
Pool based unc.	Opt.	0.71	4%	0.45	16%	0.46	17%	0.60	11%
	Max	0.76	96%	0.49	100%	0.47	100%	0.66	78%
Stream based unc.	Opt.	0.72	4%	0.41	6%	0.43	22%	0.60	12%
	Max	0.76	35%	0.47	22%	0.48	89%	0.66	100%
BADGE [11]	Opt.	0.72	12%	0.42	16%	0.38	12%	0.60	12%
	Max	0.75	55%	0.47	81%	0.47	100%	0.66	100%
CLUE [12]	Opt.	0.73	6%	0.43	13%	0.45	18%	0.61	12%
	Max	0.75	36%	0.48	91%	0.47	71%	0.67	53%
PROPOSED	Opt.	0.73	9%	0.48	13%	0.44	11%	0.61	11%
	Stop	0.73	12%	0.47	19%	0.46	28%	0.66	39%
	Max	0.75	43%	0.49	80%	0.48	83%	0.66	50%

Table 4.3: Comparison between five acquisition functions for 4 appliances from REFIT house 5: kettle, microwave, washing machine and dishwasher. The optimal points (Opt.), stopping points (Stop) and maximum performance (Max) are all included. Note that Maximum point is a point where the curves reach their maximum, which is unknown in practice and cannot be used to stop. $\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$ is the percentage of samples being labelled.

Acquisition function		Kettle		Washing M.		Dishwasher	
		F_1	$\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$	F_1	$\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$	F_1	$\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$
Pool based unc.	Opt.	0.87	4%	0.96	0%	0.75	17%
	Max	0.88	80%	0.97	56%	0.77	89%
Stream based unc.	Opt.	0.81	1%	0.96	0%	0.68	6%
	Max	0.86	17%	0.96	33%	0.76	33%
BADGE [11]	Opt.	0.84	3%	0.96	0%	0.72	18%
	Max	0.87	78%	0.97	88%	0.80	100%
CLUE [12]	Opt.	0.86	6%	0.96	0%	0.72	12%
	Max	0.87	70%	0.97	94%	0.77	47%
PRO-POSED	Opt.	0.83	1%	0.96	0%	0.65	6%
	Stop	0.86	7%	0.96	22%	0.75	17%
	Max	0.86	7%	0.96	11%	0.75	17%

Table 4.4: Comparison between five acquisition functions for 3 appliances from UK-DALE house 1: kettle, washing machine and dishwasher. The optimal points (Opt.), stopping points (Stop) and maximum performance (Max) are all included. Note that Maximum point is a point where the curves reach their maximum, which is unknown in practice and cannot be used to stop. $\frac{\lvert\boldsymbol{D}_{\text{ft}}\rvert}{\lvert\boldsymbol{D}_{\text{pool}}\rvert}$ is the percentage of samples being labelled.



(a) Kettle

(b) Microwave

(c) Washing Machine

(d) Dishwasher

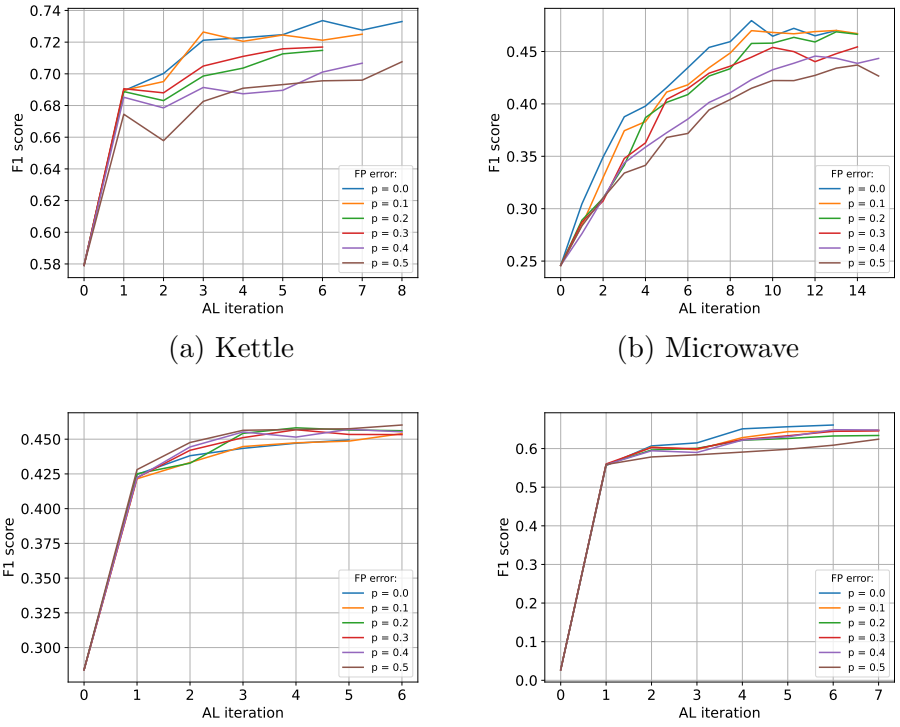
Figure 4.7: AL with simulated false negative errors in the labels for kettle (a), microwave (b), washing machine (c) and dishwasher (d) from REFIT house 5

.

correct labels, without any errors introduced. Note that the number of iterations differ across the appliances due to the proposed stopping criteria. As expected, as error probability p increases, the performance decreases - lower F_1-score is achieved. Kettle is sensitive to high levels of error, especially at later stages - it has a signature of short duration that is easily forgotten by the model if the error rate is high. Lower error rates do not impact the performance significantly. Microwave and washing machine are sensitive to this type of labelling errors even with lower error probabilities, which is reasonable since they have signatures that are already challenging to disaggregate even without any errors in labels.

Figure 4.8 shows the results when false positive errors are introduced into

labels, i.e., negative labels are set as positive. Since samples with appliance activations are more likely to be queried first as described above, the impact of false positive errors is expected to be less pronounced than the impact of false negative errors, at least in the beginning, which can be confirmed in Figure 4.8. Namely, since the dataset is already highly imbalanced in favour of sample windows without appliance activation, with false negative errors, we introduce even more negative samples, and the model starts to 'forget' the pattern it learned to recognise. On the other hand, false positive errors are likely to be introduced for samples where the aggregate signal looks as if there is appliance activation, so the model retains the ability to recognise important patterns.



(a) Kettle

(b) Microwave

(c) Washing Machine

(d) Dishwasher

Figure 4.8: AL with simulated false positive errors in the labels for kettle (a), microwave (b), washing machine (c) and dishwasher (d) from REFIT house 5.

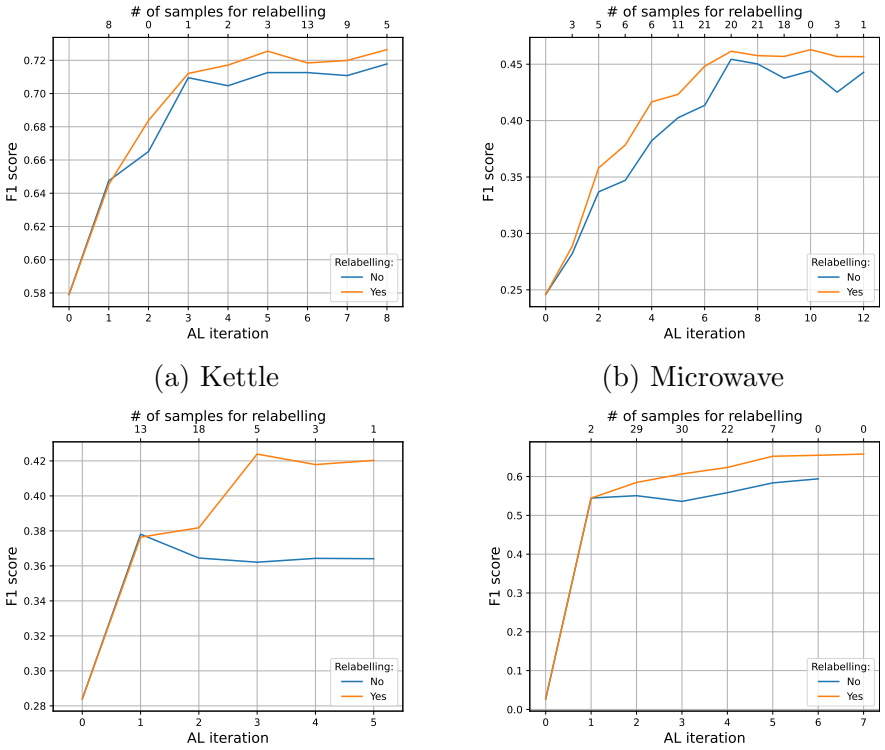
Figure 4.9 demonstrates the usefulness of the proposed re-labelling mech-

anism. Performance is compared between the case with and without re-labelling, with false negative errors occurring with the probability of 0.3. Match rate threshold T_{return} in Eq. 4.7 is heuristically set to $1e-4$ for kettle and microwave, and $5e-5$ for washing machine and dishwasher, since these appliances have longer lasting cycles and the match rate is expected to be lower even for the good predictions. The assumption is that once a sample is returned for re-labelling, a correct label is provided. Improvement in performance when using the re-labelling mechanism is observed for all four appliances, and it is most pronounced for washing machine, which is very sensitive to this type of error (see Figure 4.7c). This means that the mechanism successfully captures the samples which were wrongly labelled, and enables correcting labels by taking another look at them.

Results show that more samples are returned in AL iterations where a drop in performance is observed (e.g. iterations 6 and 7 for kettle, iterations 6-9 for microwave), indicating that the model started to adopt wrong labels, but still has not forgotten the pattern of interest, and still can detect suspicious labels. Due to the complex pattern of washing machine, the model is less confident in its predictions, and relies more and adapts to provided labels, making the predictions similar to labels, even if those are wrong. However, samples re-labelled in the beginning do improve the performance, and the improvement achieved in the beginning does not decline in later stages.

Exploiting confidence during training

Figure 4.10 shows the usefulness of the proposed modification of loss function (Eq. 4.6) to take into account confidence levels related to labels. False negative errors with probability of 0.5 are simulated. Based on the assumption that confidence level is correlated with the quality of label, two confidence levels are assumed - high confidence for samples without labelling errors and low confidence for samples containing a labelling error. The improvement in performance when using confidence levels during training compared to not using them is observable for kettle, washing machine and dishwasher from the very beginning. Even though proposed strategy improves performance for microwave and washing machine, clear convergence is not reached as with kettle and dishwasher. This is due to the complex, multi-state signatures of microwave and washing machine, as opposed to distinct patterns of kettle and dishwasher.



(a) Kettle (b) Microwave

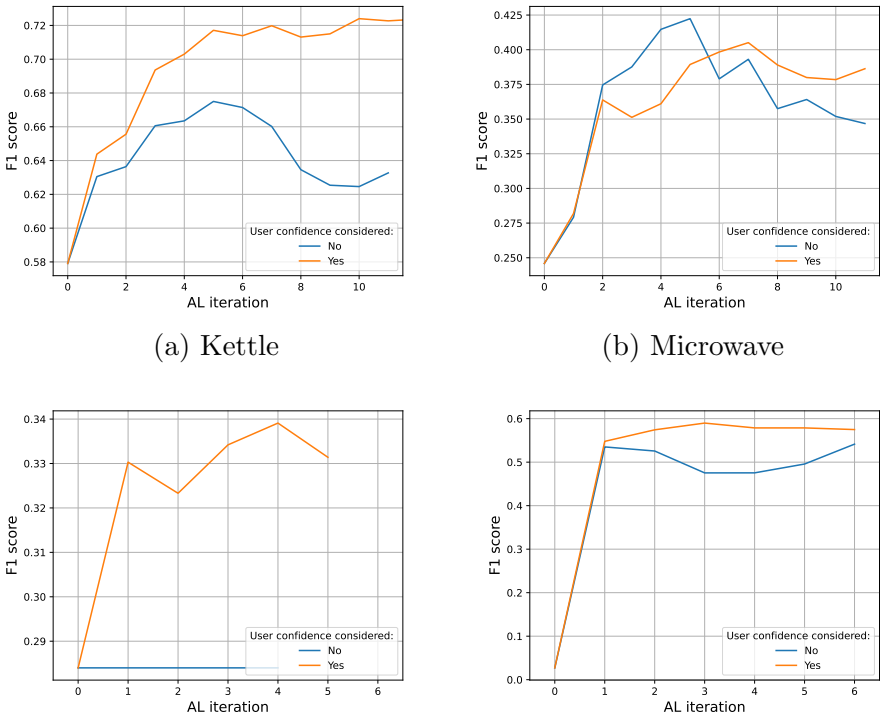
(c) Washing Machine (d) Dishwasher

Figure 4.9: The proposed AL method with and without the re-labelling mechanism for kettle (a), microwave (b), washing machine (c) and dishwasher (d) from REFIT house 5.

4.3.2 Experiment 2

In this subsection we report the results when three experts are asked to label the samples using the user interface presented in Fig. 4.4. Each expert was asked to label one or more appliances. We used the proposed acquisition function, the stopping criteria and re-labelling mechanism.

Figure 4.11 shows the results with and without using expert confidence levels. Horizontal axis represents the number of AL labelling iterations, and vertical F_1-score achieved. The blue line corresponds to the case when labels are provided by an expert familiar with NILM, but without his/her confidence levels related to each label taken into account during training (i.e., all confidence levels are set to 'high'); and the orange line corresponds to the



(a) Kettle (b) Microwave

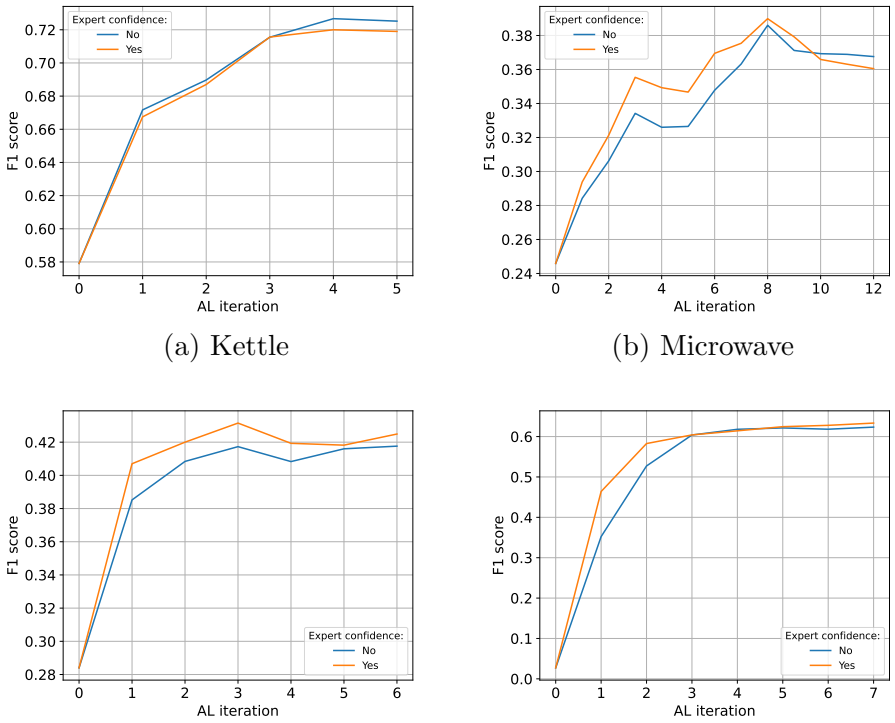
(c) Washing Machine (d) Dishwasher

Figure 4.10: AL with and without confidence taken into account during training for kettle (a), microwave (b), washing machine (c) and dishwasher (d) from REFIT house 5.

case when labels are provided by an expert, and their confidence levels are included into the loss function (Eq. 4.6) during training.

Examples of signal windows from REFIT house 5 labelled for washing machine by expert #3 and tagged with low and high confidence levels are shown in Figure 4.13, showing that more noisy samples, with not so distinct signatures, are more challenging to be labelled by naked eye.

The quality of expert-provided labels, in terms of hit, miss and false alarm, compared to the submetering ground truth is shown in Tables 4.5 and 4.6. Hit is defined as the case when the expert-provided label is overlapping with the submetering label (equivalent to TP); Miss as the case when the submetering label has an activation, but the expert-provided label does not (equivalent to FN); and False alarm as the case when the submetering label



(a) Kettle

(b) Microwave

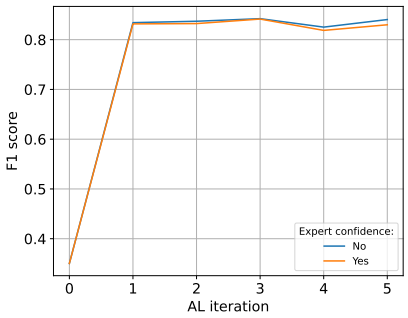
(c) Washing Machine

(d) Dishwasher

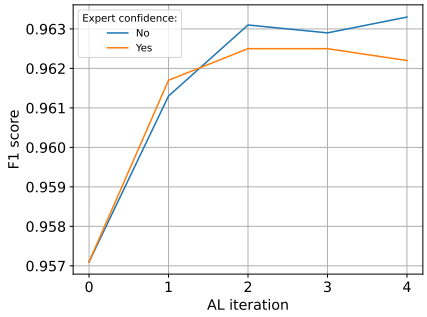
Figure 4.11: Experiment 2, REFIT house 5: Three experts asked to provide labels, where each expert labels one or two appliances. The performance curves are shown with and without expert confidence taken into account.

does not have an activation, but the expert-provided label does (equivalent to FP). In cases when there is an activation both in submetering and expert-provided label, but they do not overlap, the label falls under the Miss & False alarm category. A histogram of expert confidence levels is given next to the number of labels belonging to each of the four categories, where red denotes low confidence, yellow middle, and green high confidence levels.

For kettle from REFIT house 5 in Figure 4.11a, using confidence levels did not improve the results - the labels are already of high quality, the number of misses and false alarms is very low compared to the number of hits, which is expected since the kettle has a single state, easily recognisable signature. Moreover, the expert assigned to most of the labels high confidence, as in the no-confidence level benchmark. However, a couple of mistakes have



(a) Kettle

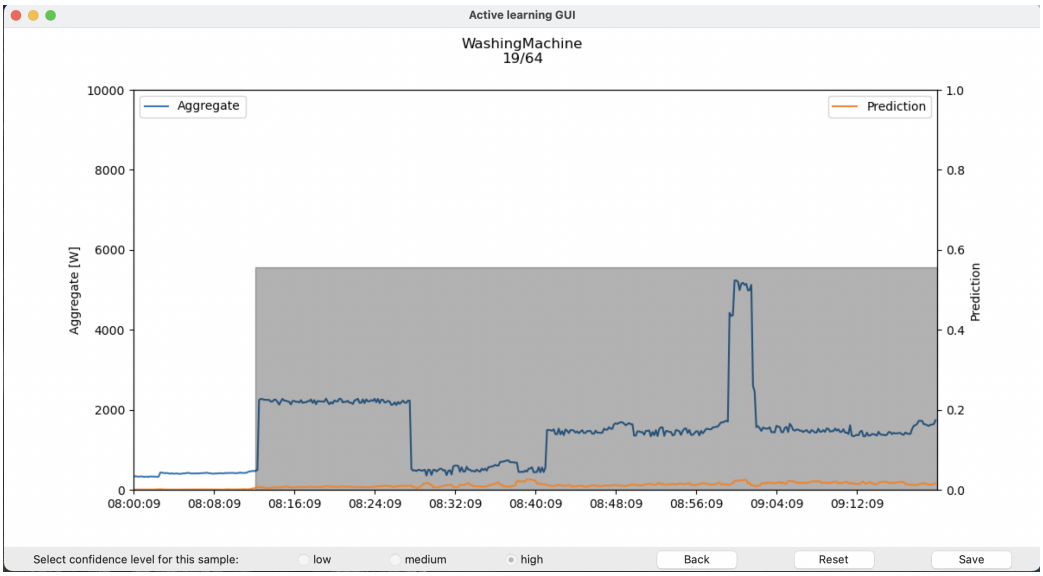


(b) Washing Machine

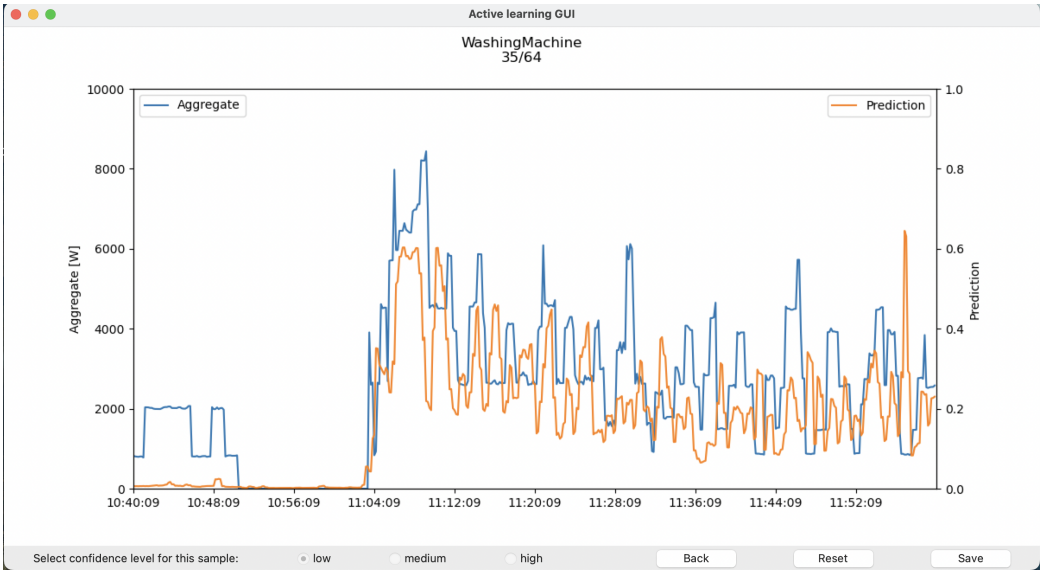
(c) Dishwasher

Figure 4.12: Experiment 2, UK-DALE house 1: Experts asked to provide labels, where each expert labels one or two appliances. The performance curves are shown with and without expert confidence taken into account.

high confidence levels, which probably caused the confidence level curve to be slightly worse than no confidence level in Figure 4.11a. The same situation is observed in UK-DALE house 1 in Figure 4.12a. For microwave (Figure 4.11b), which is a challenging appliance to label since activations are sparse and fluctuating, the power/watt level is lower compared to kettle, and there are different modes of running the appliance, expert-provided labels contain a significant number of mistakes. However, those mistakes are tagged with low confidence levels, so utilising user confidence levels did improve the results compared to the benchmark. For washing machine REFIT house 5, Figure 4.11c, which was labelled by another expert, there is a larger percentage of labelling mistakes, some of which have high confidence levels. However, there are low and mid-confidence levels among wrongly labelled samples, which



(a) High confidence - Hit



(b) Low confidence - Miss

Figure 4.13: Experiment 2: User interface showing examples of signal windows from REFIT house 5 with washing machine tagged with low and high confidence levels by expert #3.

was enough to lead to performance improvement compared to no confidence level case. For washing machine in UK-DALE house 1, Figure 4.12b, a vast majority of samples are correctly labelled, and tagged with high confidence. This causes weights to be very similar as in the case when confidence levels are not accounted for. Even though in Figure 4.12b it looks like there is a significant gap between the two curves, note that the difference is at most 0.001 in F_1-score, so performance is practically the same. For dishwasher from REFIT house 5, Figure 4.11d, labelled by another expert, the provided labels are of higher quality since they have a more distinct signature than microwave. In addition, the correct labels mostly have high confidence values, which increased the contribution of confidence exploitation, and led to minor differences between the two curves. In UK-DALE house 1, Figure 4.12c, there are very few activations among queried samples before the AL process stopped, and therefore there is almost no difference between the two curves - there are many correctly labelled negative examples tagged with high confidence levels.

	Kettle	Microwave	Washing M.	Dishwasher
Expert	#1	#3	#3	#2
Hit	113	26	46	87
Miss	36	28	25	26
False alarm	32	30	7	6
Miss & False alarm	5	2	0	0
Total # of labels	320	768	384	448

Table 4.5: Quality of expert-provided labels compared to ground truth for REFIT house 5. Red denotes low confidence, yellow middle, and green colour high confidence levels.

The main challenges encountered in this experiment are the cases when an expert assigns the same confidence value to almost all samples - then the proposed weighing of samples based on expert's confidence approaches the case when no confidence is accounted for (the vast majority of samples get the same weight). This is not the problem in datasets with low noise levels, when labels are of very high quality (for example, washing machine in UK-DALE house 1, see Table 4.6) - the most of high-confidence samples are correctly labelled; but this is a problem in very noisy datasets where there

	Kettle	Washing M.	Dishwasher
Expert	#1	#3	#3
Hit	99	78	17
Miss	12	4	6
False alarm	7	1	0
Miss & False alarm	3	0	0
Total # of labels	320	256	192

Table 4.6: Quality of expert-provided labels compared to ground truth for UK-DALE house 1. Red denotes low confidence, yellow middle, and green colour high confidence levels.

are both correct and wrong labels, but the expert is either over-confident (many wrong labels tagged by high confidence) or under-confident (many correct labels tagged with low confidence). Therefore, skill level of experts poses a limitation to this approach to some extent.

4.4 Summary

This paper proposes a human-in-the-loop AL methodology for time series data, demonstrated and evaluated for the non-intrusive load monitoring problem. Novel contributions to enable the proposed overall AL methodology comprise: design of an acquisition function based on maximum a posteriori hypothesis testing, accounting for both model uncertainty and balancing classes; a stopping criterion once optimal performance is achieved, to minimise resource-intensive labelling effort; mitigating the effect of wrong labels possibly provided by users throughout the process via two mechanisms by returning possibly wrongly labelled samples for re-labelling, and accounting for user's certainty level about provided labels, respectively.

Two experiments are conducted, applying novel AL-based approaches to the problem of time series classification of individual loads in aggregate smart meter measurements, leveraging on publicly available REFIT [52] and UK-DALE [53] datasets, and transformer-based deep learning ELECTRIcity model [70]. The first set of experiments show that the proposed acquisition function achieves similar performance to state-of-the-art methods, but with

smaller number of samples labelled due to balancing better classes and cleverly stopping when good performance is reached. Labelling effort is reduced by between 61% (in the case of dishwasher) and 88% (in the case of kettle) in REFIT house 5, and between 78% (in the case of washing machine) and 93% (in the case of kettle) in UK-DALE house 1. Furthermore, even with errors introduced throughout the labelling process, the proposed AL method enhances the model to be generalised for various profiles for the same label. The proposed re-labelling mechanism is shown to be effective in detection of mistakes during the labelling process, and offers the possibility to improve the performance by providing new labels for uncertain data samples. Finally, including confidence levels of human experts, especially in cases where samples are noisy, is beneficial as it prevents a drop in performance caused by accumulation of wrong labels. The second experiment verifies the use of proposed AL approaches in real-world scenarios, where despite unintentionally introduced errors, model performance is still boosted, especially with the use of the proposed methods for error effect mitigation.

The proposed AL approach demonstrated improved performance when pre-trained NILM models are transferred to new, unseen homes. Even when the initial performance prior to AL is poor, the proposed approach can largely improve performance by labelling a considerably small amount of data. The method can scale to many houses (hundreds, thousands) - algorithms are adjusted to each house separately - no data needs to be exported, and users (house owners) can help label their own data based on time when specific appliances are used, until the algorithms become well tuned and high performing. Considering recordings from a long period of time ensures heterogeneity of data and stability of the model. Even if circumstances in their house change (e.g., an appliance is replaced or a new high-consuming load is introduced), which impact the aggregate measurements and hence the NILM algorithm performance, the AL process can adjust the model, ensuring performance stability.

The proposed approach is demonstrated to be applicable to sensor measurements where the data being measured is fluctuating, varies across houses (domains), is noisy, and labelling is challenging. Furthermore, the very challenging nature of the load disaggregation problem is akin to the broader single source separation problem arising often from environmental sensing and therefore the method's efficacy in NILM stretches to other application domains based on solving single source separation problem from noisy time-series reading.

As some types of labels are very hard to be provided by users (for example, regression labels for the problem of load disaggregation, or strong labels for time-series windows in general), it would be worth exploring the use of Siamese networks in future work, that could be pre-trained for both regression and classification tasks at the same time, or with both strong and weak labels at the same time. Furthermore, user-provided confidence levels could be used to further train the model to learn its own confidence level. Moreover, along with model prediction, some explanation tools could be used to inform the expert of the reasoning behind the prediction to help labelling.

Chapter 5

Hybrid machine teaching for time-series classification

While previous chapters focused on AL, here we transition to MT, giving the teacher (human) more control over the training process, without the need for labelling beyond selecting several representative data samples.

As described in Section 2.4, MT can encompass a human or a machine teacher. A clear advantage of the human-as-the-teacher method under the MT paradigm is human control, including the ability to correct and interpret data labels. However, this would often require significant labelling effort. Moreover, in geoscience and environmental time-series recordings, uncertainty of human labelling is high. On the other hand, the machine-as-the-teacher method has the advantage of fast, automated labelling that does not require any domain expert input, but can lead to issues of trust and labelling error propagation. To take advantage of both methods, in this study, we propose a hybrid machine learning method that involves both human domain experts and machine (a semi-supervised Siamese deep learning network) as teacher, teaching the learner (a multi-label supervised deep learning based classifier) to classify a time-series multiple class dataset.

The human teacher annotates a few labels for the most reliable representative of each class, based on which the machine teacher, acting as "active teacher", selects and labels training examples for the learner. The human teacher monitors the learner's performance and selects a new sample representative if needed so that different examples are taught. So, with the proposed hybrid human-machine teacher model we benefit from both human and machine as teacher - domain expert's knowledge of high-level concepts is

embedded in the choice of class representatives; but the burden of evaluation of all available data, choosing and labelling all the best examples through the training stages lies on the machine teacher. Our approach is a complete black-box MT - the teacher has no access to the learner's characteristics as in [21], but in contrast to [21] does not estimate them, so the learner can be any deep learning algorithm, and the same teaching model can be used to teach multiple possible diverse learners. Even though the proposed approach has some similarities to semi-supervised learning described in Section 2.1 - a subset of the data samples (anchors) are labelled and the rest are not, in the proposed MT approach the algorithm learns gradually, and the order of samples determined based on the anchors is important.

MT, within the human-on-the-loop capability, is especially valuable to support Earth scientists to make decisions for hazard assessment, such as forecasting landslides or geothermal exploration. Landslides, which are becoming more frequent as we face the consequences of climate change, are associated with low-magnitude (micro-seismic) quakes [90], rockfalls [90] and tremor-like signals [91]. Particularly, endogenous seismicity is induced by the deformation of slow moving clay rich landslides. Slidequakes have been recorded on such unstable slopes due to the presence of material failures and shearing at the contact with the bedrock or directly within the moving mass. Locally, rockfalls can also be recorded on steep slopes while tremor-like signals may be linked to fluid transfer or transient slip. Accurate algorithms for automatic detection and classification of landslide-associated precursory events [92] are needed, so that timely action and effective management measures can be undertaken to reduce risk to life and infrastructure. Similarly, accurate micro-seismic detection that detects small fractures at sub-surface is needed to inform and guide pumping operations during geothermal exploration [93]. Unlike earthquakes, micro-seismic events are challenging to be analysed manually and algorithmically, due to the short duration and low signal-to-noise ratio in continuous seismic recordings. Indeed, it was shown recently that domain experts often miss events while cataloging large volumes of continuous data, and need the helping hand of a machine in detecting and classifying potential events, which can subsequently be verified by domain experts [8].

There has been considerable progress in the past few years in ML algorithms, achieving excellent performance for micro-seismic event monitoring - see [32] for a recent review. However, performance and usability of these methods largely depend on availability of good-quality, large, labelled

datasets. Since the data can only be annotated by domain experts (who are usually not ML experts), despite availability of seismic recordings, creation of large labelled datasets, necessary for training of AI algorithms is an expensive and time-consuming task.

In this chapter, we propose a hybrid MT approach for trustworthy seismogram classification. We propose an approach where a domain expert, i.e., a geoscientist, acts as a human teacher by controlling the content of the training dataset. This is achieved with relatively minimal effort, by only choosing representative data samples for each class, referred to as *anchors*.

These anchors serve two purposes: (i) to curate the training dataset by a machine teacher, by incrementally adding samples that are closest to the anchors, mimicking the way humans learn - starting from simple, more obvious examples and moving towards more challenging ones. Thus, our approach falls under the sequential MT paradigm [7]; (ii) to automatically label the training dataset without requesting a domain expert to label all training samples. The classifier neural network, acting as a learner, is then trained incrementally using the training samples provided. After setting the anchors, the domain expert can monitor the classifier performance and eventually intervene and change the anchors as and when needed.

We leverage upon the multi-label classifier CNN of [33], which operates on seismometer measurements in the time domain, and is proven to achieve state-of-the-art performance if trained with a large labelled dataset. We use the same publicly available dataset as [33], namely the Résif [75] dataset containing seismic records from the Super-Sauze landslide to support reproducibility of our research. The latter is characterised as a slow-moving clay-rich landslide, where analysis of microseismicity is a challenging task because the signals are of low magnitude ($ML < 1$), low amplitude (< 10000nm/s), and are generally highly attenuated at short distances (< 200m) [76].

5.1 Methodology

This section describes the proposed hybrid MT framework for seismic event classification, shown in Figure 5.1. The task is to classify samples from an unlabelled time-series dataset, $\boldsymbol{D}_s$, referred to as *sampling set*, using (any) state-of-the-art deep learning-based model m, such as [33, 8, 94, 67].

The proposed solution is to pass a small set of samples (i.e., events to classified) to a domain expert, who is assumed not to have any machine learning

expertise. The domain expert, acting as the *human teacher*, identifies distinct classes in the provided set of samples and selects an example for each class, he/she is confident to label. The domain expert-labelled samples (one per class) are then fed into the *machine teacher*, implemented as a Siamese deep neural network, which labels all remaining samples present in the sampling set. The Siamese network-labelled samples are then ranked based on their distance to the expert-labelled samples, and gradually, batch by batch, starting from the top ranked samples, they are fed into the classification algorithm (i.e., *the learner*) for training.

Thus, the proposed hybrid MT framework consists of three steps that are performed iteratively: (i) A human teacher, domain expert, chooses a set of samples (called *anchors*) from $\boldsymbol{D}_s$, each representing one distinct class. (ii) a machine teacher (Siamese deep neural network) ranks and labels training samples based on the domain expert-provided anchors, and (iii) a learner, i.e., a seismic event classification algorithm, is trained based on the provided labelled samples. In the remainder of this section, we describe each of these steps, one by one.

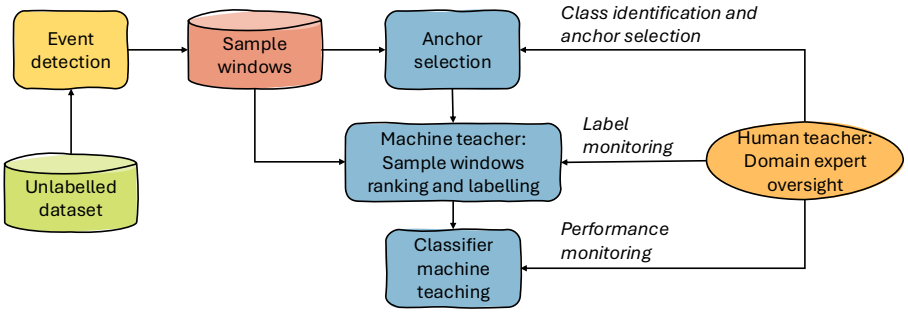


Figure 5.1: The proposed hybrid MT framework.

5.1.1 Human teacher: Anchor selection

The human teacher, i.e., the domain expert, starts the learning process by selecting distinct anchors from the unlabelled sampling set $\boldsymbol{D}_s$. In particular, $\boldsymbol{D}_s$ is a set of micro-seismic events obtained from raw seismometer measurements using STA/LTA algorithm as described in [76]. The extracted events are shown to the domain expert who identifies the initial number of classes present in the dataset, and selects class representatives, called anchors. Note that new classes can be added during the iterative learning process. The expert can either set the first sample of each class as anchor, or can quickly scan through the dataset and pick the most reliable sample as anchor.

Visualization approaches, such as the one proposed in [95] can be used to make more accurate and faster selections. The best anchor choice can be made using the cosine distance values between pairs of samples in the sampling set $\boldsymbol{D}_s$. Based on these distances, all samples can be clustered using, for example, an agglomerative clustering algorithm. The events are then sorted by cluster labels and visualized in a heatmap as in Figure 5.2. The darkest values in the heatmap indicate lower cosine distances, and the lighter values indicate greater distances. Dark triangular areas on the vertical axis indicate parts of the sampling set that are good anchor candidates for the label on the horizontal axis. However, light horizontal/vertical lines inside the dark triangles correspond to samples that have a high distance to events in the same cluster, and dark lines in the lighter part of the heatmap correspond to samples that have a low distance to samples outside their clusters. This can happen due to interclass similarity or high noise levels, and these events should not be selected as anchors.

Another way to assist human expert in anchor selection, is estimating Signal-to-Noise Ratio (SNR) defined in (5.1) [66] to filter detected events and present only high SNR ones to the expert, reducing the number of samples the expert needs to scan through, and helping them select reliable, high SNR anchors containing representative, clean events:

$$SNR = \frac{\sqrt{\sum_{i=1}^{l}(s_i)^2}}{\sqrt{\sum_{i=1}^{l}(n_i)^2}}, \tag{5.1}$$

where l is the number of samples in a window, s and n are, signal and noise windows, respectively, obtained after denoising with band-pass filtering.

Each anchor, a_c, selected by the expert, represents one class c. Let $\boldsymbol{C}$ be

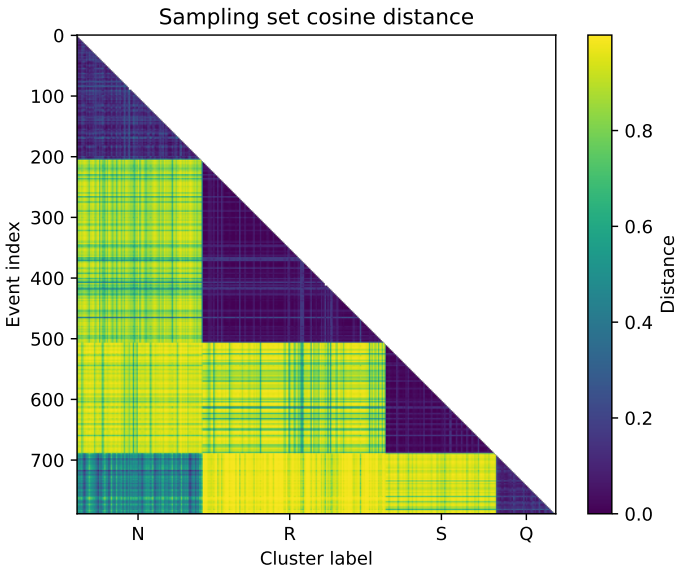


Figure 5.2: A visualization tool to help with anchor selection.

the set of all classes identified in this way. The task is then to automatically label all remaining samples into $|\boldsymbol{C}|$ classes, each represented by a domain expert-selected anchor.

5.1.2 Machine teacher: Sample ranking and labelling

The next step is labelling the sampling set $\boldsymbol{D}_s$ by the machine teacher, implemented by a Siamese neural network. Siamese networks are based on the concept of similarity learning, performed by comparing two network inputs and calculating the difference between their encoded network representations (embeddings).

Let s_i, s_j be two input samples from $\boldsymbol{D}_s$. Siamese networks perform metric or similarity learning, i.e., they learn a function $m_s(s_i, s_j)$ that compares two input samples (one input being the anchor a_c and the other is each unlabelled sample in $\boldsymbol{D}_s$). Specifically, the networks first perform identical transformation of the two input samples and then apply a distance metric to estimate similarity, i.e., $m_s(s_i, s_j) = d(\tau(s_i), \tau(s_j))$, where function d is the distance metric and the transform τ is applied to both input data samples

for identical representation learning. The function τ is usually implemented as a deep CNN.

The main advantage of Siamese networks compared to other metric learning approaches, lies in its ability to perform jointly feature representation and metric learning. A recent survey [96] of Siamese networks highlights the benefits these networks provide and their ability to learn with unlabelled data, by comparing directly embeddings rather than relying on labels.

The Siamese networks comprise: feature extractor, comparison head, and decision-making head. Typically, the feature extractor (that implements function τ) contains two identical branches used to learn the best feature representation for the two input samples (s_i and s_j). The comparison head applies a distance metric d to compare similarity between the two embeddings, while in the final steps the decision-making head performs classification by comparing the output of the comparison step with a pre-set threshold.

Specifically, our Siamese neural network m_s computes the distance of each sample s in the sampling set $\boldsymbol{D}_s$, to each of the anchors, according to:

$$m_s(a_c, s) = d(\tau(a_c), \tau(s)) = 1 - \frac{\tau(a_c) \cdot \tau(s)}{\|\tau(a_c)\| \cdot \|\tau(s)\|}, \tag{5.2}$$

where $m_s(a_c, s)$ denotes Siamese neural network output for the two inputs - anchor a_c and a sample from the sampling set s. Vector embeddings of the two inputs are denoted as $\tau(a_c)$ and $\tau(s)$. Function d represents cosine distance. $\tau(a_c) \cdot \tau(s)$ is the dot product of the two embeddings, and $\|\tau(a_c)\|$ and $\|\tau(s)\|$ are their Euclidean norms. Note that cosine similarity $\frac{\tau(a_c)\cdot\tau(s)}{\|\tau(a_c)\|\cdot\|\tau(s)\|}$ ranges from -1 to 1, so $d(\tau(a_c), \tau(s))$ ranges from 0 to 2.

The samples are ranked based on the calculated distances, and only top ranked samples (i.e., those with least distance to the anchors), are included in the training set $\boldsymbol{D}_t$ together with their estimated class membership defined as:

$$l(s) = \arg\min_{c \in \boldsymbol{C}} m_s(a_c, s). \tag{5.3}$$

This way, the classification model m learns gradually, starting with the simplest (most confident) and moving towards more complex concepts. Namely, the samples that are the closest to anchors are chosen in the first stage, and then more samples that are further from anchors are added incrementally at later stages. See Figure 5.3 for an example of sample selection with two classes and batch size of 8. An equal number of top-ranked samples are taken

per anchor, i.e., $n = batchsize/\mid \boldsymbol{C} \mid$ at each stage, ensuring that all classes are represented at early stages of MT even if there is class imbalance in the dataset (see example in Figure 5.3). The selection is made class by class, and there is no repetition among the selected samples, i.e., if sample s is the top-n ranked sample for multiple anchors, its label is calculated using (5.3), and the sample is removed from the ranked lists of all the classes $c \neq l(s)$ before the top n selection is made (in Algorithm 2, for loop in line 16 runs class by class, and selection for one class is removed from the sampling set in line 19 before the selection for the next class is made).

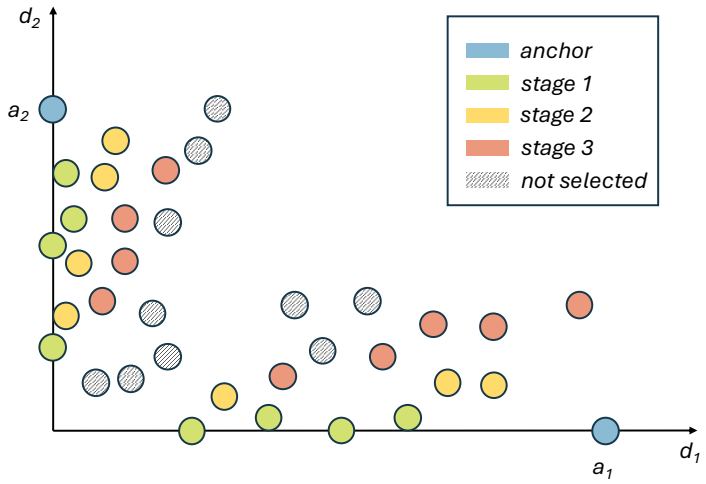


Figure 5.3: Sample selection example - a 2-dimensional feature space; there are 2 classes, i.e., two anchors (blue; a_1 and a_2); 8 samples are selected per stage. Green samples are selected in the first stage, yellow in the second, red in the third and gray are not selected.

5.1.3 Learner: Iterative learning

As described previously, machine-labelled sample windows are fed into the learner, i.e., classifier CNN, for training in stages. At each stage, new samples (closest to the anchors) are added to the training set, and the classifier model

is fine-tuned using the new, extended training set. At each stage, the classifier CNN's learning rate is decreased by 5%, so that the best quality samples have the highest impact on the model weights, and the more complex ones, which have a higher chance of being wrongly labelled, have less impact on the model weights.

The human teacher oversees the process - having an option to monitor labels created by the machine teacher as well as the performance of the classifier CNN, and select new anchors. The human expert will automatically be asked to change the anchors once all the distances of the samples from the sampling set $\boldsymbol{D}_s$ to the current anchors become greater than the threshold T defined at the beginning of the process as:

$$T = \frac{1}{\mid \boldsymbol{D}_s \mid} \sum_{s \in \boldsymbol{D}_s} \min_{c \in \boldsymbol{C}} m_s(a_c, s) \tag{5.4}$$

The rational behind this selection of T, is that once the samples become far from the initial anchors (the distance is above the average value across all samples), their machine-teacher set labels are unreliable, and could deteriorate the learning process. Hence, the human teacher is asked to provide new anchors from the reaming set of unlabelled samples.

The overall framework described above is presented in Algorithm 2.

The training can continue until all available samples from the sampling pool $\boldsymbol{D}_s$ are labelled and included in the training set, or until a stopping criterion is met - e.g., the learner performance does not improve for certain number of training stages, or after all remaining samples in the sampling pool $\boldsymbol{D}_s$ are further from anchors than a threshold.

The proposed approach differs from Active Teacher model defined in [2], where the Active Teacher is a machine that does not examine the learner directly, but rather checks the learner's performance to determine its status. In the proposed approach, the hybrid teacher consists of a human teacher who steers training of the learner by setting anchors, and is aided by a machine teacher who ranks samples to feed them to the learner in stages, and labels them based on provided anchors. The human teacher is in control of checking learner's performance instead of the machine teacher. Since in the proposed approach, the learner is a complete black-box to the teacher, the hybrid human-machine teacher can be used with any deep learning model as a learner, such as [33, 8, 94, 67], which is in contrast to [97, 21].

Algorithm 2 Hybrid MT framework

Variables:

$\boldsymbol{A} \leftarrow \emptyset$ ▷ Set of anchors to be chosen by human teacher
$\boldsymbol{C} \leftarrow \emptyset$ ▷ Set of classes identified by human teacher
m ▷ the learner - classifier
m_s ▷ machine teacher
$batch\_size$ ▷ Batch size for training of m
$\boldsymbol{D}_s$ ▷ Sampling set, unlabelled
$\boldsymbol{D}_t \leftarrow \emptyset$ ▷ Training set
$\boldsymbol{D} \leftarrow \emptyset$ ▷ Selected samples
$T \leftarrow 0$ ▷ Threshold for replacing anchors based on (5.4)

Procedure:

$\boldsymbol{A} \leftarrow \{a_c\}$, *selected by human teacher*
$T = \frac{1}{|\boldsymbol{D}_s|} \sum_{s \in \boldsymbol{D}_s} \min_{c \in \boldsymbol{C}} m_s(a_c, s)$
while $\boldsymbol{D}_s \neq \emptyset$ or early stopping criteria reached **do**
  for $c \in \boldsymbol{C}$ **do**
    $\boldsymbol{D}_s = sort(\boldsymbol{D}_s, by\ m_s(a_c, \boldsymbol{D}_s))$ given by (5.2)
    $\boldsymbol{D} = \boldsymbol{D}_s[: batch\_size\ /\ |\ \boldsymbol{C}\ |]$
    $\boldsymbol{D}_s = \boldsymbol{D}_s \setminus \boldsymbol{D}$
    $\boldsymbol{D}^{labelled} = (\boldsymbol{D}, \arg\min_{c \in \boldsymbol{C}}(m_s(a_c, \boldsymbol{D})))$
    $\boldsymbol{D}_t = \boldsymbol{D}_t \cup \boldsymbol{D}^{labelled}$
  end for
  fine-tune$(m, \boldsymbol{D}_t)$

  if $\exists c, \min_{s \in \boldsymbol{D}_s} m_s(a_c, s) \geq T$ **then**
    $\boldsymbol{A} \leftarrow \{a_c\}$, *selected by human teacher*
    $\boldsymbol{C} \leftarrow \{c, \forall a_c \in \boldsymbol{A}\}$
    $T = \frac{1}{|\boldsymbol{D}_s|} \sum_{s \in \boldsymbol{D}_s} \min_{c \in \boldsymbol{C}} m_s(a_c, s)$
  end if
end while

5.2 Experimental design

This section describes the dataset used to showcase the proposed method (including selected anchors), the machine teacher implementation, i.e., the Siamese neural network, the seismic event classification model used as a learner within the framework, and evaluation metrics used.

5.2.1 Dataset

An open access dataset from the Résif Seismological Data Portal, recorded by the French Landslide Observatory Observatoire Multidisciplinaire des Instabilités de Versants (OMIV) [75], described in Section 2.7.2, is used in this paper.

Original data are resampled to 100Hz to comply with pre-trained, off-the-shelf Siamese neural network used within the framework. Window length used is 10 seconds, thus dimensions of one sample are 1000×3. For each event, the sample window starts one second before the start of an event (detected using STA/LTA algorithm as described in [76]) e.g., if the timestamp is 2013-10-23 15:34:24, then the window starts at 2013-10-23 15:34:23. Sample windows are split as follows: 60% for the sampling set $\boldsymbol{D}_s$ - used without labels, 10% for the validation set $\boldsymbol{D}_{val}$, and 30% for the test set $\boldsymbol{D}_{test}$ for each class, chronologically. The number of events per class is given in Table 5.1.

Class	Total	Test
Earthquake (S)	335	113
Quake (Q)	207	69
Rockfall (R)	351	116
Noise (N)	302	105

Table 5.1: Dataset structure - number of samples per class.

At the start of the experiments, a domain expert was asked to select a set of anchors after being presented with unlabelled sample windows from the sampling set $\boldsymbol{D}_s$. The timestamps of events selected by micro-seismic expert for this set are as follows: 2013-10-23 15:34:24.340 for earthquake (S); 2013-11-09 00:54:45.060 for quake (Q); 2014-11-22 16:48:19.390 for rockfall (R); and 2013-11-03 03:43:12.500 for noise (N). We refer to this set of anchors as $\boldsymbol{A}_1$.

5.2.2 Machine teacher implementation: Siamese neural network

A pretrained Siamese neural network [95], composed of two-branch fully convolutional feature extractor and a comparison head is used to curate the training set for the classifier CNN and automatically label it. Each feature extractor branch processes one input of size 1000×3, and comparison head computes Cosine distance between obtained encodings, as given by (5.2). That is, for two signals that are the same, the output is 0, and for two very distinct signals, the output should be close to 1. The architecture is shown in Figure 5.4. Convolutional layers use 'relu' activation function, while dense layers use 'sigmoid' activation function. The Siamese neural network is pretrained using the Résif dataset, with the same train-test split as in [95] to avoid data leakage. Training is performed with a 5-fold cross-validation. Adam optimizer is used with a learning rate of $5e-4$, and the batch size used is 256 [95]. This neural network shows robust generalisation capabilities, as demonstrated in [92], where it is trained with the Résif dataset, and successfully transferred to the Hollin Hill dataset. Full training details of the used Siamese neural network can be found in [95].

The Siamese neural network is used to determine the distance between each selected anchor and every sample from the sampling set, based on (5.2), where the anchor is the input to the Siamese neural network and a sample from the sampling set is the test input. Cosine distance is not sensitive to magnitude distances, and hence is a reliable metric for the application being considered [95]. Then, the training set is gradually expanded with a number of samples that are closest to each of the anchors, and those samples are given labels based on (5.3).

A plot of the distances of sample windows from the test set to the four anchors $\boldsymbol{A}_1$, calculated by Siamese neural network using (5.2), is shown in Figure 5.5. Each subplot corresponds to one class from the test set, and each colour corresponds to one anchor. Median distance between each of the anchors and all test sample windows is shown in Table 5.2, grouped by labels of the test sample windows (each row of the table corresponds to sample windows from one class). It can be seen from these distance plots that the sample windows belonging to the earthquake class (S) are close to the anchor representing that class, and they are far from anchors representing other classes (Figure 5.5, bottom histogram - S; Table 5.2 - median S-S distance 0.06). Similar observation holds for the rockfall sample windows (R),

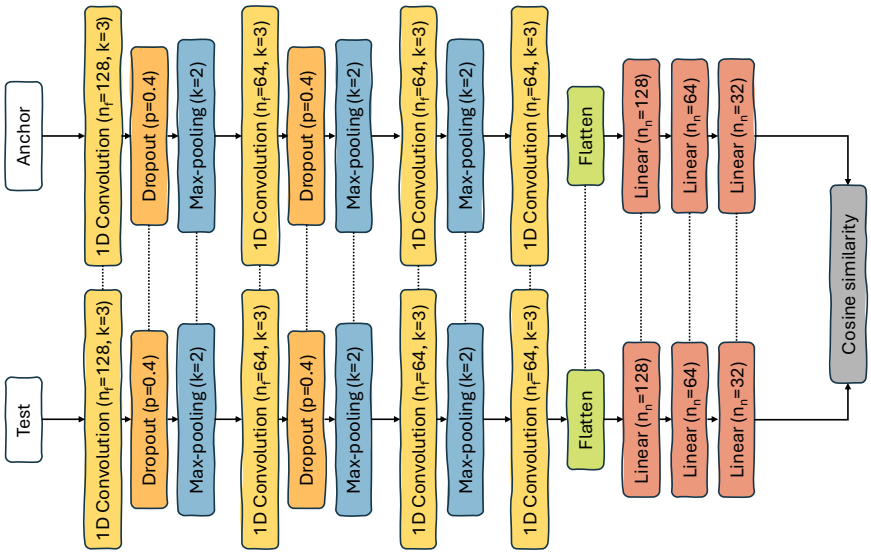


Figure 5.4: Siamese neural network. n_f - number of filters in convolutional layers, n_n - number of neurons in linear layers; k - kernel size; p - dropout rate.

except that some of them are close to the anchor representing the earthquake class. On the other hand, the quake (Q) and noise (N) class samples are not always close to their respected anchors. Due to the short duration of quakes, and false alarms during detection phase, they can be harder to distinguish from noise [66]. Indeed, some of the noise sample windows are very close to the anchor representing the quake class (Figure 5.5, second plot - Q), which is expected to impact the performance not only for these two classes, but also for the earthquake class, since quakes and earthquakes can appear similar, especially in the case of low-magnitude earthquakes, attenuated at short distance, as present in slow-moving clay-rich landslides [76](Figure 5.5, top plot S - some quake sample windows are close to the anchor representing earthquakes).

Labels generated according to (5.3) for the test set using the Siamese neural network and the four anchors have weighted F_1 score of 0.70 compared to the catalogue (S:0.79, Q:0.51, R:0.81, N:0.58). Therefore, automatically generated labels are imperfect, but the proposed MT approach is expected to mitigate the effects of labelling mistakes. For comparison, if the classifier

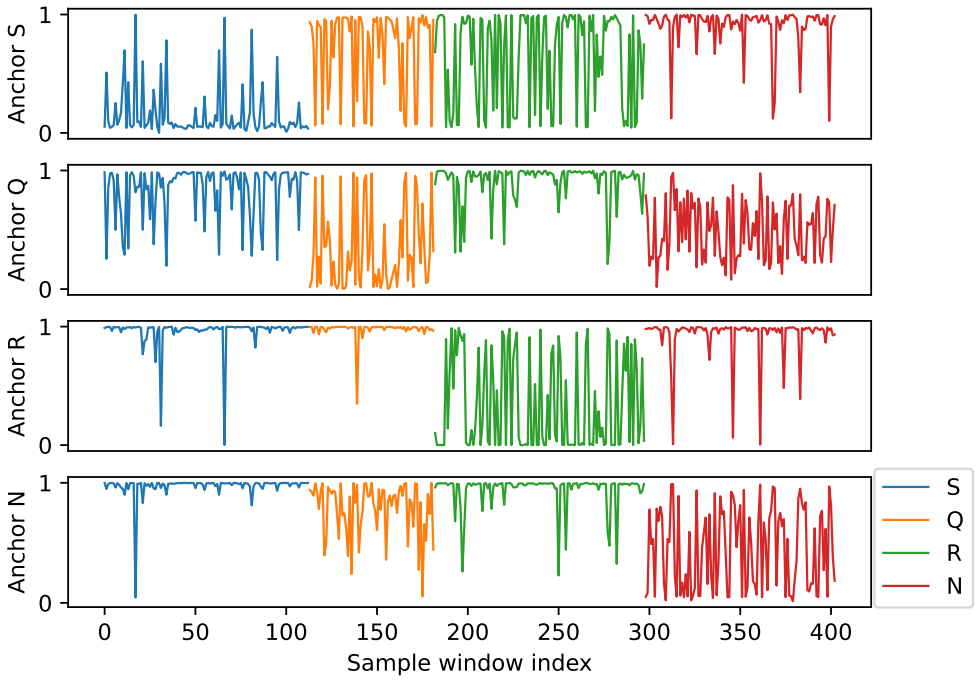


Figure 5.5: Histograms of Siamese distances of test samples from anchors $\boldsymbol{A}_1$ representing each class. Note that y-axis is in log scale.

	Anchor			
Label	S	Q	R	N
S	0.064353	0.943853	0.992817	0.995988
Q	0.907087	0.216816	0.995869	0.885996
R	0.864959	0.975084	0.069447	0.991390
N	0.958279	0.447776	0.983292	0.489018

Table 5.2: Median Siamese distance of the test set sample windows from anchors $\boldsymbol{A}_1$ representing each class.

CNN is trained using a classical "black-box" machine learning approach, with the whole sampling set at once (not in stages) and the catalogue labels, it reaches F_1 score of 0.86 (S:0.91, Q:0.76, R:0.90, N:0.82).

5.2.3 Learner implementation: Seismic event classification model

Within the proposed framework, a CNN for seismic event classification inspired by [33] is used as learner. Originally, the network was designed to process 6-channel input windows (from one 3-channel and 3 single-channel sensors), 10 seconds long, sampled at 250Hz (which gives an input dimension of 2500×6). Since an off-the-shelf pre-trained Siamese neural network with 3-channel (from a 3-channel sensor only) and 10 seconds long input at a sampling frequency of 100Hz is used as machine teacher, CNN for seismic event classification is adjusted to process inputs of the same dimensions, 1000×3. The output of the classifier CNN is a 4×1 vector, containing probabilities of the input sample belonging to each of the 4 classes. The architecture of the classifier CNN is shown in Figure 5.6.

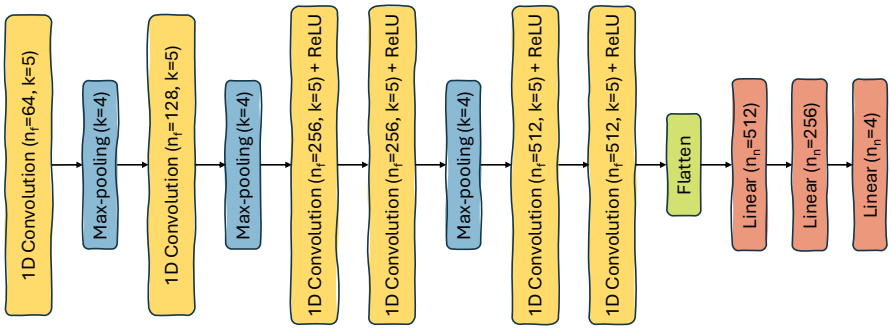


Figure 5.6: Seismic signal classification model [33]. n_f - number of filters in convolutional layers, n_n - number of neurons in linear layers; k - kernel size.

All experiments start with an untrained learner model. At each stage, the learner, classifier CNN, is trained for a maximum of 15 epochs, keeping the model that performs the best on the validation set, and after each stage, the learning rate is reduced by 5%. Hyper-parameters used for model training are summarized in Table 5.3. Classification performance of the CNN model for the seismic event classification is measured using F_1 score and weighted F_1 as described in Section 2.8, Equation 2.4.

Parameter	Value
Batch size	16
Learning rate	7e-4
Learning rate decay	5%
Epochs per iteration	max 15

Table 5.3: Hyper-parameters for the CNN model training

5.3 Results & Discussion

In this section, we present and discuss experimental results. We compare complexity and classification performance of the following schemes:

1. Random selection: The training samples are chosen randomly - there is no ranking based on anchors; anchors only serve for automatic labelling, and they are set by the domain expert as good representatives of classes present (anchor set $\boldsymbol{A}_1$, see Section 5.2.1). This scheme is used to demonstrate that the order of samples from which the classifier CNN gradually learns is important.

2. AL: The learner, i.e., the classifier CNN, is in control of the learning process. The anchors set by the domain expert ($\boldsymbol{A}_1$, see Section 5.2.1) are used for labelling only, and sample windows are selected at each training stage by the classifier CNN based on the least confidence when classifying samples from the sampling set. Classifier confidence is calculated as predicted class probability: $\max_{c \in \boldsymbol{C}} m(s), s \in \boldsymbol{D}_s$. The idea behind this is that the samples with the least confidence bring the most information to model training, and if they are included in the training set, the classifier performance is expected to improve rapidly. The classifier CNN is initialised in one iteration as in the proposed scheme, using the anchors and the machine teacher (Siamese neural network) to ensure it acquires some knowledge to start with, for fair comparison to other schemes. This scheme is to show that using teacher-driven learning (i.e., ranking and selecting samples based on teacher's confidence) is essential in challenging datasets, and that AL is less robust to wrong labels compared to the proposed hybrid MT approach.

3. Hybrid MT (the proposed scheme): In this scheme, anchors that are set by the domain expert serve both for sample windows ranking and

labelling, as described in Section 5.1. This scheme demonstrates the efficiency of the proposed methodology - the classifier CNN can learn fundamental concepts with imperfect labels if the training dataset is well organised - samples ranked according to similarity to class representative samples, i.e., according to label confidence.

5.3.1 Classification performance

The average performance measure in terms of F_1 score per class of the schemes described above is shown in Table 5.4, while Table 5.5 shows confusion matrices.

Figure 5.7 shows performance of the Siamese network only as the accuracy of the generated labels after each stage (cumulative, taking into account all labels generated at each stage) in terms of F_1 score for each class (colored), and weighted F_1 score (black). Labels for quake (Q) and noise (N) samples are more inaccurate than for earthquake (S) and rockfall (R) samples, which was expected based on the distance of anchors from the test set sample windows as discussed earlier and shown in Figure 5.5. The results are inline with those reported in [95] where a semi-supervised Siamese-based network with Short-Time Fourier Transform (STFT) input is used on the same dataset, and the averaged reported F1-scores were 0.81, 0.88, 0.63, and 0.7, for R, S, Q, and N classes, respectively.

Scheme	S	Q	R	N	Weighted F_1	Avg F_1
1 - Random	0.69	0.57	0.70	0.56	0.64	0.63
2 - AL	0.82	0.55	0.80	0.61	0.71	0.70
3 - MT	0.86	0.61	0.87	0.70	0.78	0.76

Table 5.4: F_1 score of the classifier CNN model for the experimental schemes with automatic labelling - random teacher (Scheme 1), AL (Scheme 2), and the proposed MT (Scheme 3).

When the Random scheme is used, the accuracy of the labels generated by the Siamese neural network quickly converges to the final F_1 score and no further improvement occurs (Figure 5.7a), which is expected since the samples are selected randomly. This machine-teacher labelling performance, is reflected into the learner's performance, as can be seen from Table 5.4 as this scheme has the poorest average performance among all the schemes.

Label	S	Q	R	N
S	101	7	0	5
Q	20	42	1	6
R	36	1	65	14
N	22	29	3	51

(a) Random - Scheme 1

Label	S	Q	R	N
S	107	5	0	1
Q	12	46	3	8
R	24	5	83	4
N	5	42	6	52

(b) AL - Scheme 2

Label	S	Q	R	N
S	99	9	1	4
Q	10	44	3	12
R	4	1	98	13
N	3	22	8	72

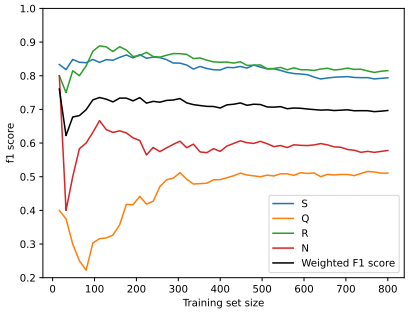
(c) MT - Scheme 3

Table 5.5: Confusion matrices of Random, AL and MT schemes.

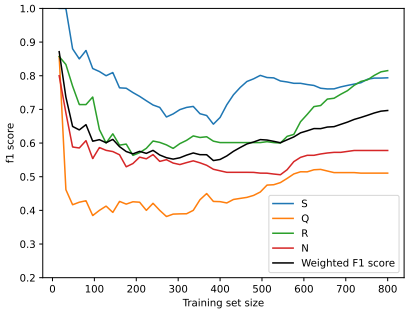
In the corresponding confusion matrix (Table 5.5a), there is a bias towards S class, due to earthquake events occurring more frequently compared to quake and noise events (see Table 5.1); hence, when randomly choosing samples to include in the training set, earthquake samples are more dominant. Furthermore, some of the rockfall samples have a low Siamese distance from the anchor representing earthquakes, as discussed before and shown in Figure 5.5. These results demonstrate that using randomly selected samples to train the classifier is ineffective, since many wrongly labelled samples are used for training from the start, preventing further improvements.

The AL scheme outperforms the Random scheme in terms of the classifier F_1 score (Tables 5.4 and 5.5b). However, the accuracy of the labels generated by Siamese neural network, decreases first in the early stages of training, and then increases and reaches the final label F_1 score (Figure 5.7b). This is due to the fact, samples that were challenging for the learner (i.e., the classifier CNN) to predict (i.e., that are predicted with the least confidence), were also challenging for the teacher (i.e., Siamese neural network) to classify - therefore, samples included in the training set early, contain high amount of labelling errors. Thus, the AL approach is not very robust to potentially noisy labels.

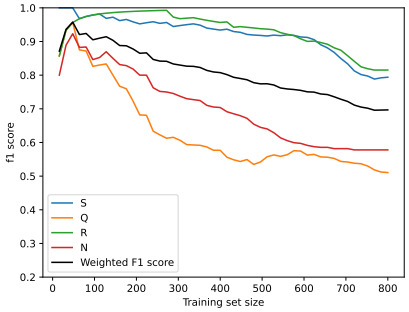
The proposed MT Scheme 3 yields a better overall result than the AL scheme - the quality of selected anchors, especially those representing earth-



(a) Random - Scheme 1



(b) AL - Scheme 2



(c) MT - Scheme 3(i)

Figure 5.7: Cumulative F_1 score of labels generated by the Siamese neural network when samples are selected randomly (a), by the classifier CNN (b), and by the teacher (c).

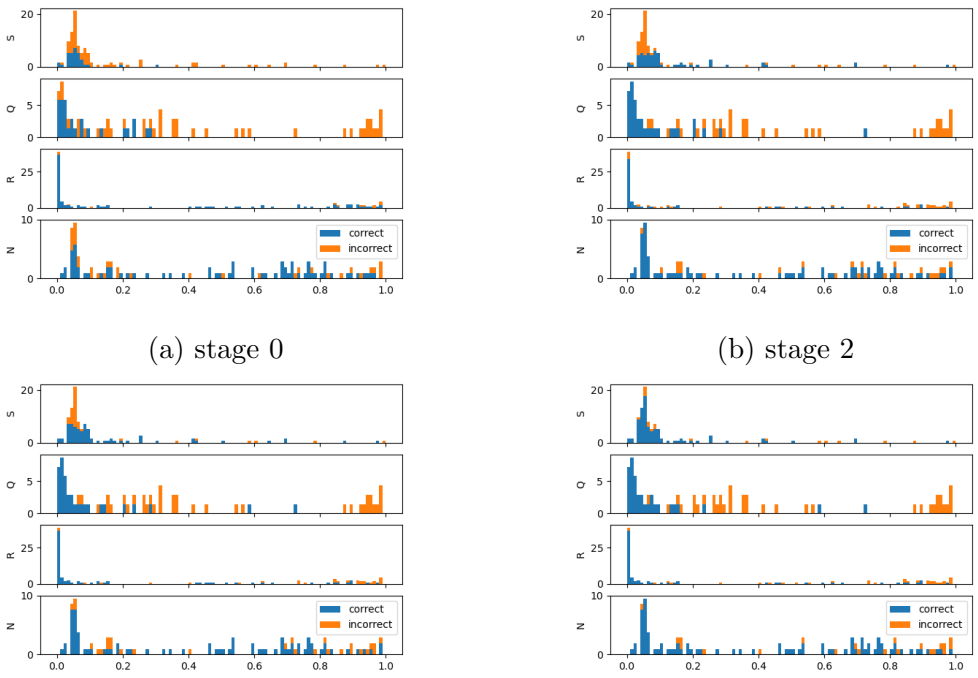
quake and rockfall classes, contributed to training a well-performing CNN classifier for these classes. In this scheme, the best quality labels are provided for all classes in the beginning, and as the process progresses, and samples further from anchors (and with lower SNR) are included in training, the label quality deteriorates (Figure 5.7c). But, since good quality and correctly labelled samples are presented to the classifier CNN at the beginning of the training, with a higher learning rate, they pose the basis of the classifier CNNs knowledge, and this secures a high model performance, despite unclear and potentially wrongly labelled samples at later stages. Due to the closeness of noise samples to the anchor representing the quake class, misclassification between these two classes can be observed (Table 5.5c). Figure 5.8 shows a progression of model performance with the proposed MT Scheme

3, in terms of histograms of Siamese distances between correctly (blue) and incorrectly (orange) classified samples from each class (note that the histograms are stacked, not overlapped) and the anchor representing that class, for the several training stages (0, 2, 5 and 10) when the most of the model performance improvement occurs. As the training progresses, the classifier CNN improves its classification performance, especially for the samples close to the anchors; learning is the slowest for the quake and noise classes, inline with observations about closeness of the quake anchor to noise samples from Figure 5.5 and Table 5.2. At Stage 10 already, most samples close to the respective anchors are classified correctly for all classes, and for some classes, such as the earthquake class, even samples away from the anchors are mostly correctly classified.

The MT results are similar to those reported in [98] (Figure 5.4), where, for the same dataset, the highest class-average F_1 score of 72% was reported for unsupervised classification, and performance improvement of up to 10% required 10% of data labels. As per Table 5.4, Scheme 3 achieves average F_1 score of 0.76.

5.3.2 Complexity

The used Siamese neural network has 1,052,320 trainable parameters, while the classifier CNN has 3,944,836 trainable parameters. Running times of all the schemes are measured on an Apple M1 Max chip, with 32GB RAM, and results from the first iteration, (i.e., stage) including ranking of samples belonging to $\boldsymbol{D}_s$, labelling of one batch of sample windows, and classifier CNN training for 15 epochs with one batch of sample windows, with sampling set $\boldsymbol{D}_s$ size of 789 and batch size of 16 are summarized in Table 5.6. Note that the Random scheme does not involve sample ranking. AL scheme requires running the CNN model on the sampling set $\boldsymbol{D}_s$ in the sample ranking step, while the proposed MT scheme involves running the Siamese neural network on $\boldsymbol{D}_s$ against each of the anchors in the sample ranking step, but only at the beginning of the process and every time the anchors are changed. There is no need to re-sort $\boldsymbol{D}_s$ if anchors stay the same. The Random and AL schemes require running the Siamese neural network within the labelling step, however, for the porposed MT scheme, labelling is included in the sample windows ranking step since the Siamese model is already run at that step. In conclusion, the proposed MT scheme brings performance improvement at additional sample ranking running time compared to the Random



(a) stage 0

(b) stage 2

(c) stage 5

(d) stage 10

Figure 5.8: Histograms of Siamese distances of correctly (blue) and incorrectly (orange) classified test sample windows by the CNN classifier from the anchor from $\boldsymbol{A}_1$ representing the corresponding class.

scheme, but at less running time compared to the AL scheme. The proposed scheme can rank and label around 256 samples per second and therefore can be used in real-time applications even if the sampling set $\boldsymbol{D}_s$ is constantly updated with new samples. Training of the classifier CNN consumes the largest amount of time among all steps within one iteration (i.e., stage) and is therefore the biggest bottleneck.

Scheme	Sample ranking	Labelling	Training
Random	–	66ms	22.225s
AL	3.100s	66ms	22.225s
MT	3.072s	–	22.225s

Table 5.6: Running times of the three experimental schemes

As the sampling set can grow rapidly when the framework is used in real time, leading to increased sample ranking time, the sampling set can be implemented as a circular buffer. The buffer would maintain a fixed size, and if the limit is reached, the oldest samples would be overwritten as the new ones arrive.

Note that, the above comparison does not include anchor selection and possible re-selection, since all three methods require this identical step. Anchor selection can be likened to labelling, but through the proposed approach, the human expert would only need to label/select one anchor or representative signal per class. This should not take long for a human expert. Please refer to Section 5.1.1, where we discuss how to facilitate anchor selection.

5.3.3 Ablation study 1: Robustness to anchor selection

To test the robustness of the proposed approach, we ask domain expert to choose two more sets of anchors - another set of good-quality anchors, $\boldsymbol{A}_2$, with the following events included: 2015-06-20 20:11:56.140 for earthquake (S), 2013-11-13 17:12:23.180 for quake (Q), 2014-11-10 19:07:00.830 for rockfall (R), and 2013-10-17 13:40:05.570 for noise (N); and a set of noisy anchors, $\boldsymbol{A}_3$, with the following events included: 2013-11-09 03:34:27.620 for earthquake (S), 2013-11-02 03:01:03.480 for quake (Q), 2014-11-10 23:19:58.660 for rockfall (R), and 2013-10-20 20:03:17.180 for noise (N). Please note that noise in the noisy set of anchors is not artificially introduced; it is a natural characteristic of signals. Then, we test the robustness of the proposed framework to anchor selection with the following anchor settings:

1. Using the same noisy anchors ($\boldsymbol{A}_3$) throughout the whole process, without updating;
2. Noisy anchors ($\boldsymbol{A}_3$) are updated with the good quality anchors ($\boldsymbol{A}_1$) as described in Section 5.1, when the conditions of (5.4) are met;
3. Good-quality anchors ($\boldsymbol{A}_1$) are updated with another set of good-quality anchors ($\boldsymbol{A}_2$); and
4. Using average distance from good quality anchors ($\boldsymbol{A}_1$) and noisy anchors ($\boldsymbol{A}_3$).

Results in terms of F_1 score are presented in Table 5.7 and confusion matrices are shown in Table 5.8. Figure 5.9 shows performance of the Siamese

network only as the accuracy of the generated labels after each stage (cumulative, taking into account all labels generated at each stage) in terms of F1 score for each class (colored) and weighted F1 score (black), as in Section 5.3.1.

Anchor setting	S	Q	R	N	Weighted F_1	Avg F_1
1	0.78	0.47	0.82	0.64	0.70	0.68
2	0.80	0.53	0.86	0.63	0.73	0.71
3	0.84	0.56	0.82	0.72	0.76	0.74
4	0.78	0.50	0.84	0.54	0.69	0.67

Table 5.7: Results of anchor sensitivity analysis - F_1 score.

Label	S	Q	R	N
S	99	9	0	5
Q	13	31	1	24
R	20	0	84	12
N	9	24	3	69

(a) Anchor setting 1

Label	S	Q	R	N
S	101	9	0	3
Q	15	39	3	12
R	12	1	93	10
N	11	29	5	60

(b) Anchor setting 2

Label	S	Q	R	N
S	96	7	0	10
Q	10	32	2	25
R	6	1	84	25
N	4	6	2	93

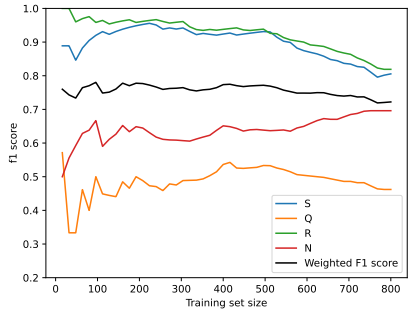
(c) Anchor setting 3

Label	S	Q	R	N
S	97	12	1	3
Q	14	43	2	10
R	15	3	88	10
N	9	46	3	47

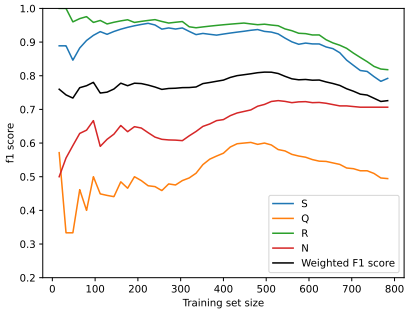
(d) Anchor setting 4

Table 5.8: Results of anchor sensitivity analysis - confusion matrices.

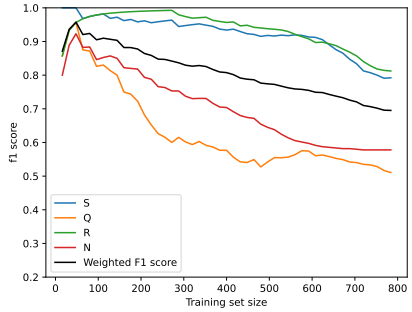
With Anchor setting 1, where noisy anchors are used throughout the process, the generated labels (Figure 5.9a) are not as accurate in the early learning stages as those with MT (Scheme 3) in Section 5.3.1, where clear, good quality anchors are used. Although noisy anchors do not necessarily mean that the overall F_1 score of all generated labels will be significantly worse than in the case of clear anchors (see the final values in Figures 5.7c and 5.9a), they lead the training process by introducing noisy samples instead



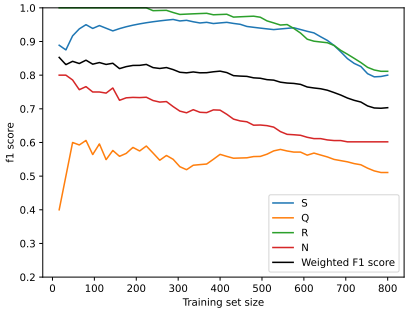
(a) Anchor setting 1



(b) Anchor setting 2



(c) Anchor setting 3



(d) Anchor setting 4

Figure 5.9: Anchor sensitivity analysis: Cumulative F_1 score of labels generated by the Siamese neural network with anchor settings 1 (a), 2 (b), 3(c) and 4(d).

of clear ones into the training set first. Thus, there are more labelling errors in the early stages of the process. Consequently, it is harder for the classifier CNN to extract characteristic features of events, and the final performance of the classifier trained with noisy anchors is worse compared to the case when clear anchors are used, as shown in Table 5.4.

Anchor setting 2 results show that, even if the teaching process starts with unreliable anchors, the final performance of the classifier can be improved by replacing them with better, more representative anchors - see Table 5.7 for a comparison between Anchor settings 1 and 2. Anchors for N, Q and R classes are replaced at training set sizes 304, 320, and 656, respectively. Figure 5.9b shows the label accuracy improvement after noisy anchors are replaced with good-quality ones.

Anchor setting 3 demonstrates the robustness of the proposed approach if the starting good quality anchors are replaced with another set of anchors - performance is very similar to MT (Scheme 3) results reported in Section 5.3.1, as shown in Table 5.7, and also the label accuracy (Figure 5.9c) is similar to Figure 5.7c, implying that performance is stable as long as good class representatives are set as anchors.

Anchor setting 4 results show that noisy anchors can negatively affect the performance even if they are used together with good quality anchors and the distance from them is averaged for every sample and for every class. The label accuracy in Figure 5.9d is better compared to Anchor setting 1 (Figure 5.9a) in the beginning, but worse for N class in the end. Labels are less accurate compared to the results presented in Section 5.3.1 and Anchor setting 3 where good quality anchors are used. This again emphasizes the importance of choosing anchors that are good class representatives, and domain experts can undoubtedly do this successfully.

According to the confusion matrices in Table 5.8, quake class (Q) is very sensitive to the choice of anchor - the performance for Q class results presented in Section 5.3.1 is better compared to other anchor settings, which is expected since these events are characterised by low magnitude and short duration. Further, as discussed in Section 5.2.2, Siamese neural network does not distinguish well quakes and noise even with good quality anchors, and this is reflected in the classifier CNN performance in Table 5.8. When using noisy anchors, rockfalls (R) are mistakenly classified as earthquakes (S). Samples belonging to earthquake class are not confused with other classes even if noisy anchors are used, indicating that this class is the most robust, due to its distinctive waveform.

5.3.4 Ablation study 2: Cosine distance threshold for anchor update

To evaluate the sensitivity to the cosine distance threshold for anchor changes, we conduct additional experiments. In these experiments, the anchors are adjusted when the closest data sample reaches distances of 0.2, 0.5, and 0.8 from the anchor, and according to (5.4). The results are shown in Table 5.9.

Low threshold values cause change of anchors very soon after the start of the process, while higher threshold values lead to a very late or no change of anchors. If the anchors are not good enough, changing them too late

limits the benefits of the change, since the learning rate decreases during the process, and the model weights are already heavily impacted by the old anchors.

Good threshold values depend on the data, and even though manually set low threshold values perform similar to the one set using (5.4) in this case, the latter provides a data-driven approach and hence is more favorable - it follows the data and eliminates the need for manual threshold setting.

Threshold	S	Q	R	N	Weighted F_1	Avg F_1
0.2	0.83	0.48	0.83	0.67	0.73	0.70
0.5	0.79	0.40	0.81	0.64	0.69	0.66
0.8	0.79	0.43	0.82	0.67	0.70	0.68
(5.4)	0.80	0.53	0.86	0.63	0.73	0.71

Table 5.9: Results of ablation study on cosine distance threshold for anchor update.

5.3.5 Transferability testing: different dataset and different learner

Dataset: To test the transferability of the proposed framework to another, unseen environment, we test it with Stanford Earthquake Dataset (STEAD) [34], a large-scale global data set of local earthquake and non-earthquake signals recorded by seismic instruments. The dataset contains local earthquake waveforms, recorded within 350 km of earthquakes, and seismic noise waveforms that do not contain earthquake signals.

Chunks 0 and 1 of the dataset are used, containing seismic noise signals and local earthquake signals, respectively. Each data sample window in the dataset is a 3-channel seismogram, 1 min long at a sampling frequency of 100 Hz (each sample window has dimensions 6000×3). Each sample window is associated with metadata providing information related to measuring instruments and detected event (in case of local earthquake samples), including P and S wave arrival, coda-end (i.e., the late-arriving, low-amplitude tail of seismic waves on a seismogram, which consists of energy that has been scattered multiple times by Earth's heterogeneities) and SNR. Since the Siamese neural network used in this study processes inputs of dimensions 1000×3, which corresponds to the sample window length of 10 seconds (s), local earthquake

data samples are filtered to discard events longer than 9 s. Only samples where coda-end comes at most 9 s after P wave arrival are kept, and new windows are created by cutting the original windows from 1 s before P wave arrival, to 9 s after P wave arrival. For samples containing seismic noise, the middle of the original window is kept, i.e., the original window is cut from 25 s to 35 s. After filtering, there are 61,323 local earthquake sample windows and 235,426 seismic noise sample windows. To speed up the testing, this filtered dataset is subsampled with a factor of 0.2, preserving the ratio of the number of local earthquakes and seismic noise sample windows, resulting in 47,085 seismic noise and 12,265 local earthquake sample windows. The query pool / validation / test split used is 60/10/30%.

Anchor: Since there are only two classes in the dataset - local earthquake and seismic noise, only one anchor is selected for the local earthquake class. Then, the Siamese neural network calculates cosine distance between a sample and the anchor, and compares the result to a threshold - if it is lower than the threshold, the sample is classified as a local earthquake, and if it is greater, the sample is classified as seismic noise. The anchor is selected using the SNR values present in the metadata for each event sample - the one with a high SNR values across 3 channels is selected, with trace name B014.PB_20150929033005_EV. The threshold for classification by the Siamese model is set at 0.1, heuristically, based on the histogram of the cosine distances for all query pool data samples, as shown in Figure 5.10. The F_1 score of labels generated automatically using Siamese neural network and the selected anchor on the test set is 0.85. A number of noise samples appears very close to the local earthquake anchor, possibly indicating missed events. Since there is only one type of seismic event in this dataset, in the sample selection step during MT, one half of batch is selected as sample windows closest to the anchor, and the other half is selected from the query pool randomly.

Learner implementation: In this transferability testing scenario, instead of [33], we use the ConvNetQuake deep neural network, a popular open-source model used for seismogram classification [99]. It is a lightweight model composed of eight convolutional layers followed by a linear layer, whose architecture is shown in Figure 5.11. Training hyperparameters are summarized in Table 5.10.

Results: MT is run for 20 iterations in this transferability testing scenario, since the query pool is very large. The ConvNetQuake model trained using the proposed framework reaches a F_1 score of 0.91, with the highest

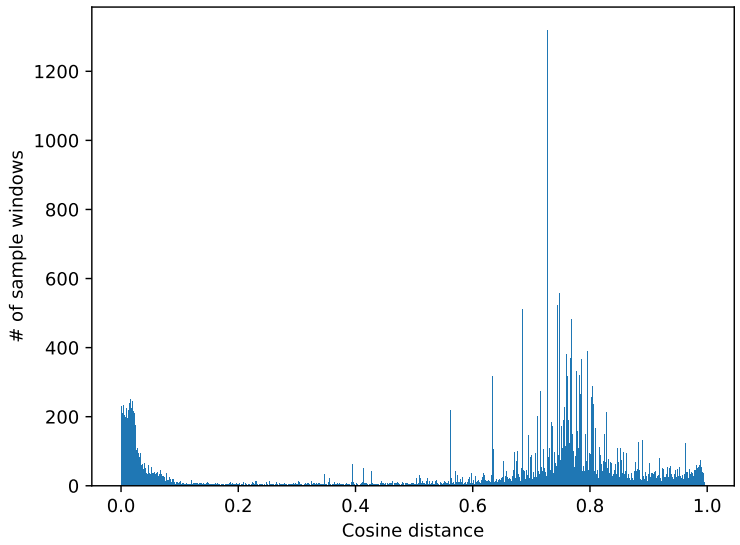


Figure 5.10: Histogram of cosine distances of query pool samples from the anchor, STEAD dataset.

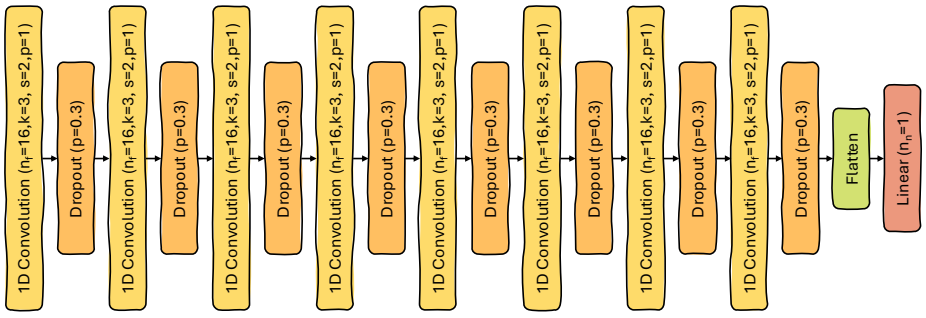


Figure 5.11: Learner architecture for transferability testing scenario.

improvement at the beginning, and the performance remains steady in later iterations. As shown in Figure 5.10, the cosine distances of samples classified as local earthquake class from the specified anchor are predominantly centered around 0, indicating that these samples are closely aligned with the anchor point, and they do not bring much information as MT progresses. The confusion matrix is presented in Table 5.11.

Parameter	Value
Batch size	16
Learning rate	7e-4
Learning rate decay	5%
Epochs per iteration	max 15

Table 5.10: Hyper-parameters for the ConvNetQuake model training

Label	Noise	Local earthquake
Noise	13922	204
Local earthquake	468	3212

Table 5.11: Confusion matrix of the ConvNetQuake model trained with the proposed MT framework.

The above presented results demonstrate the transferability of the proposed MT framework. Furthermore, the observed high F_1 score and the low confusion rate for local earthquake samples suggest that the model effectively distinguishes between local earthquake events and noise. The consistent performance across iterations reinforces the robustness of the framework, indicating that early training iterations yield the most informative updates. However, as the cosine distances from the anchor are not very diverse, future iterations may benefit from tuning or strategic sampling to capture additional diversity in the query pool. The results also demonstrate that the proposed methodology is applicable to different networks, including lightweight models such as ConvNetQuake.

5.3.6 Summary of the results

In summary, the proposed MT framework with a hybrid human-machine teacher outperforms Random and AL schemes for training a micro-seismic event classifier CNN in a setup where labelled data is not available, and annotations are generated automatically based on domain expert input (labelling only one sample per class). The following key observations are made:

1. MT is essential to rank training samples based on their label reliability, estimated by the distance from the anchors, i.e., by label reliability.

2. AL, where a learner queries samples of least confidence, is not robust

when training set is noisy and hard to label.

3. Human input is critical to select good anchors, since performance for some classes is sensitive to the anchor quality. However, if the initial poor anchor choice is corrected, classifier performance quickly improves. The framework is shown to be robust to changing anchors throughout the process as long as they are good class representatives.

4. The schemes show high performance even with only four labelled samples with a help of machine teacher.

5. Hybrid MT outperforms AL despite lower computational complexity.

5.4 Summary

This chapter proposes a novel hybrid MT framework for seismic event classification with expert oversight. The teacher is of hybrid nature - a human teacher (i.e., a domain expert) is aided by a machine teacher. The human teacher steers the learning process by setting anchor signals representing each class present in the dataset, based on which the machine teacher ranks the training samples for learning in stages, and labels them. It is demonstrated that the proposed hybrid MT methodology is effective for training a classifier CNN for seismic event classification. Expert's workload is minimal, and training in stages teaches the classifier CNN gradually, managing the effect of potential errors in automatic labelling. The approach is validated in several experimental scenarios, and it outperformed the random teacher and AL approaches, demonstrating that embedding human oversight and domain knowledge beyond labelling is essential in developing accurate and trustworthy deep learning algorithms.

Limitations of the proposed approach include difficulties in selection of high-quality anchor signals, due the high volume of data that needs scanning, and low signal-to-noise ratio in the recordings, where visualisation approaches as in [95] can be beneficial. Future work includes investigation on efficient strategies to aid the human teacher in selection of high-quality anchors, and to embed explainable AI techniques in classifier's performance monitoring to further inform the human teacher of the training progress, as well as refining sampling strategy in cases when all data samples are very close to the anchors

and ranking by distance does not offer substantial advantages for improving model training.

Chapter 6

Conclusion

6.1 Summary

The thesis explores challenges related to utilising labelled data to efficiently train AI models. Moreover, it tackles the problem of model deterioration over time. Additionally, it incorporates human agency and oversight as one of the fundamental trustworthy AI principles. To this end, several approaches for integrating human agency and oversight, while minimising the amount of data that needs labelling without compromising the performance, are examined. These range from active learning (AL), which involves experts participating directly in the labeling process, to machine teaching (MT), where an expert is asked only to select representative samples to guide and monitor the learning process.

In Chapter 3, oracle-based AL approaches are explored for the NILM and micro-seismic event detection problems. Various acquisition functions, transferability, different training modes, and use of weak labels are considered. The labelling effort is reduced by 85-95% in the case of the NILM problem with strong labels, 82.6-98.5% in the case od NILM with weak labels, and 83% in the case of micro-seismic event detection. This chapter answers the first research question of the thesis - *'Can oracle-based deep active learning be useful for efficient training and transfer of AI algorithms applied to time-series data classification?'*, demonstrating effectiveness of AL approaches to reduce the amount of labelled data needed for the two problems, and is the basis upon which the rest of the thesis is built.

Chapter 4 introduces the first human-in-the-loop AL framework for the

NILM problem. The effects of human-introduced errors are studied and strategies for their mitigation are proposed. This is important since the challenges of real-world implementation of AL are addressed - there is no oracle and the AL process is imperfect. In these real-world settings, model transferability is significantly improved with labelling effort reduced by 61-93%. This chapter answers research questions 2 and 3 - *'When and how to optimally stop the active learning process? Can an acquisition function with an inbuilt stopping mechanism be designed to be used within an active learning framework?'* and *'When and how to optimally stop the active learning process? Can an acquisition function with an inbuilt stopping mechanism be designed to be used within an active learning framework?'*. An acquisition function based on hypothesis testing is designed to timely stop the active learning process when additional labelling is unlikely to bring a significant performance improvement. Experiments indicate that the proposed strategies designed to mitigate effects of imperfect labels (based on expert confidence and detection of possibly wrong labels) are effective in an AL framework for NILM.

In Chapter 5, a MT framework for micro-seismic event detection is designed, with a hybrid human-machine teacher, giving the human more meaningful task of steering the training process of the algorithm by anchor selection, and using the machine part of the teacher for the task of labelling and training set curation. This gives human a higher degree of control over the algorithm life cycle while minimizing the amount of time required from an expert. The proposed approach outperformed the random teacher and AL scenarios (F_1 score 0.64 and 0.71, respectively) achieving an F_1 score of 0.78. This chapter answers the last research question of the thesis - *'Can expert's knowledge be used more efficiently through machine teaching with a hybrid human-machine teacher, i.e., can experts be included in the process with a higher level of control over the process, and with less labelling effort?'*. The proposed MT framework enables an expert to guide and oversee the training of a deep learning model while labelling only a few representative data samples, reaching performance that is on par with models trained using fully labelled datasets.

The work in this thesis applied to NILM resonates with the UN Sustainable Development Goals 7 (Affordable and Clean Energy) and 12 (Responsible Consumption and Production) by providing users with a clear and easy-to-understand summary of their energy expenses, helping them see when and how much energy they use, as well as whether it's from renewable or non-renewable source. Work applied to micro-seismic event classification

resonates with SDG 13 (Climate Action) by strengthening resilience and adaptive capacity to climate- related hazards and natural disasters such as landslides.

Whereas the methods proposed in this thesis are applied to the challenges of NILM and micro-seismic event classification, they are versatile and can be extended to other time-series signals characterized by large volumes of data and difficulty of labelling. In scenarios where human agency and oversight are essential, these approaches can provide valuable solutions.

While the thesis presents innovative frameworks for active learning (AL) and machine teaching (MT), there are some limitations. Taking part in model training requires knowledge about how AI algorithm training works, and lay users may not know which samples bring the most information to model training, as this might differ from what is intuitive in some cases. Also, the proposed methods rely on experts' skill level - in Chapter 4, experts can be under- or overconfident, making it harder to estimate label accuracy and take it into account in training. In Chapter 5, experts need to choose good quality anchors for the MT process, otherwise performance is sacrificed. Additionally, the proposed methods for engaging domain experts and end users in algorithm maintenance may face challenges related to user adoption.

6.2 Future work

Future research can focus on several key areas to improve AL and MT frameworks for time-series data developed in this thesis.

One possible future research direction is the integration of explainable AI (XAI) methods into both AL and MT frameworks, enabling more transparent and trustworthy interactions between human experts and models in time-series domains such as NILM and microseismic analysis. For instance, experts could be presented with saliency maps during the AL query selection. These maps assign an importance score to each temporal point in the input signal (e.g., power consumption waveforms in NILM or seismic waveforms in microseismic data), highlighting regions where the model focuses its attention for appliance disaggregation or event detection. This visualization not only reveals where the model "looks" but also aids in diagnosing misclassifications, such as confusing similar appliance signatures in NILM. Shapley values offer another complementary XAI approach, quantifying the marginal contribution of each signal point to the model's prediction. In a microseis-

mic context, for example, Shapley values could attribute higher weights to P-wave arrivals or high-amplitude bursts, helping experts verify if the model correctly prioritizes seismically relevant phases over noise. Beyond these, Layer-wise Relevance Propagation (LRP) could be adapted to propagate relevance scores backward through the network, producing heatmaps that are particularly interpretable for recurrent or convolutional architectures common in time-series modeling. For AL, this might involve selecting queries not only based on uncertainty but also on regions of high relevance disagreement between model and expert, fostering targeted teaching in MT. Counterfactual explanations represent yet another avenue: generating 'what-if' perturbations to the time-series input (e.g., minimally altering a seismic trace to flip an event classification) to illustrate decision boundaries. Integrating these XAI techniques—potentially in an interactive dashboard—would provide deeper insights into the model's reasoning processes, such as identifying systematic biases in handling variable-length events or noisy baselines. This transparency can empower users to better understand the model's strengths (e.g., robust handling of periodic patterns) and weaknesses (e.g., vulnerability to sensor drift), thereby informing more precise decisions during AL querying or MT guidance. Ultimately, such enhancements could lead to accelerated convergence, reduced labeling efforts, and superior model performance in real-world deployments.

Another area of exploration is the development of more engaging methods for involving domain experts and end users in the ongoing maintenance of algorithms. This could include creating collaborative platforms that facilitate real-time feedback and input from users, with intuitive, user-friendly interfaces, ensuring that the models remain relevant and effective in dynamic real-world environments. Engaging users in this manner not only enhances the model's adaptability but also fosters a sense of ownership and investment in the technology.

Another promising direction for future work is the adaptation of AL strategies for deployment on edge devices, where computational resources, energy consumption, and latency are severely constrained. This involves developing lightweight query selection mechanisms that operate efficiently in low-power environments, such as embedded systems or IoT sensors processing time-series data in real time. For instance, edge-based AL could prioritize uncertainty sampling or diversity-based criteria using quantized models or approximate inference to minimize communication with central servers while maintaining labeling efficiency. Additionally, exploring feder-

ated AL paradigms where edge devices collaboratively select informative samples without sharing raw data could enhance privacy and reduce bandwidth requirements. Such advancements would enable the proposed frameworks to support on-device continual learning, making them viable for resource-limited applications like remote environmental monitoring, or NILM, particularly through integration with edge-centric NILM methods that perform real-time appliance disaggregation directly on smart meters or gateway devices [100].

Furthermore, future work will investigate the scalability of the proposed frameworks to various households within NILM environments, as well as across different domains and applications involving other types of time-series signals. Understanding how these approaches can be generalised or tailored to such varied contexts will be crucial for their widespread adoption.

Bibliography

[1] E. Commission, C. Directorate-General for Communications Networks, and Technology, *Ethics guidelines for trustworthy AI*. Publications Office, 2019.

[2] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal, "Human-in-the-loop machine learning: a state of the art," *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3005–3054, 2023.

[3] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[4] P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang *et al.*, "Machine teaching: A new paradigm for building machine learning systems," *arXiv preprint arXiv:1707.06742*, 2017.

[5] T. Todic, L. Stankovic, V. Stankovic, and J. Shi, "Quantification of dairy farm energy consumption to support the transition to sustainable farming," in *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, Jul. 2022, pp. 368–373.

[6] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI magazine*, vol. 35, no. 4, pp. 105–120, 2014.

[7] X. Zhu, A. Singla, S. Zilles, and A. N. Rafferty, "An overview of machine teaching," *arXiv preprint arXiv:1801.05927*, 2018.

[8] J. Jiang, D. Murray, V. Stankovic, L. Stankovic, C. Hibert, S. Pytharouli, and J.-P. Malet, "A human-on-the-loop approach for labelling seismic recordings from landslide site via a multi-class deep-learning based classification model," *Science of Remote Sensing*, p. 100189, 2025.

[9] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[10] T. Todic, V. Stankovic, and L. Stankovic, "An active learning framework for the low-frequency non-intrusive load monitoring problem," *Applied Energy*, vol. 341, p. 121078, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261923004427

[11] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds," *arXiv preprint arXiv:1906.03671*, 2019.

[12] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman, "Active domain adaptation via clustering uncertainty-weighted embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8505–8514.

[13] D. Kothandaraman, S. Shekhar, A. Sancheti, M. Ghuhan, T. Shukla, and D. Manocha, "Salad: Source-free active label-agnostic domain adaptation for classification, segmentation and detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 382–391.

[14] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical Image Analysis*, vol. 71, p. 102062, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841521001080

[15] Z. Zhang, E. Strubell, and E. Hovy, "A survey of active learning for natural language processing," *arXiv preprint arXiv:2210.10109*, 2022.

[16] A. Kirsch, J. Van Amersfoort, and Y. Gal, "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning," *Advances in neural information processing systems*, vol. 32, 2019.

[17] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," *arXiv preprint arXiv:1708.00489*, 2017.

[18] T. Ueno, H. Ishibashi, H. Hino, and K. Ono, "Automated stopping criterion for spectral measurements with active learning," *npj Computational Materials*, vol. 7, no. 1, p. 139, 2021.

[19] J. Zhu, H. Wang, E. Hovy, and M. Ma, "Confidence-based stopping criteria for active learning for data annotation," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 6, no. 3, pp. 1–24, 2010.

[20] M. Bloodgood and K. Vijay-Shanker, "A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping," *arXiv preprint arXiv:1409.5165*, 2014.

[21] W. Liu, B. Dai, X. Li, Z. Liu, J. Rehg, and L. Song, "Towards black-box iterative machine teaching," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3141–3149.

[22] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[23] 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.

[24] P. Huber, A. Calatroni, A. Rumsch, and A. Paice, "Review on deep neural networks applied to low-frequency nilm," *Energies*, vol. 14, no. 9, p. 2390, 2021.

[25] B. Zhao, M. Ye, L. Stankovic, and V. Stankovic, "Non-intrusive load disaggregation solutions for very low-rate smart meter data," *Applied Energy*, vol. 268, p. 114949, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030626192030461X

[26] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive appliance load monitoring using low-resolution smart meter data," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2014, pp. 535–540.

[27] R. Mollel, L. Stankovic, and V. Stankovic, "Using explainability tools to inform nilm algorithm performance: a decision tree approach," in *6th International Workshop on Non-Intrusive Load Monitoring, NILM2022*, Oct. 2022, pp. 1–5. [Online]. Available: http://NILM2022

[28] G.-F. Angelis, C. Timplalexis, S. Krinidis, D. Ioannidis, and D. Tzovaras, "Nilm applications: Literature review of learning approaches, recent developments and challenges," *Energy and Buildings*, vol. 261, p. 111951, 2022.

[29] P. Kumar and A. R. Abhyankar, "A time efficient factorial hidden markov model-based approach for non-intrusive load monitoring," *IEEE Transactions on smart Grid*, vol. 14, no. 5, pp. 3627–3639, 2023.

[30] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1739–1747, 2018.

[31] B. Zhao, K. He, L. Stankovic, and V. Stankovic, "Improving event-based non-intrusive load monitoring using graph signal processing," *IEEE Access*, vol. 6, pp. 53 944–53 959, 2018.

[32] D. Anikiev, C. Birnie, U. bin Waheed, T. Alkhalifah, C. Gu, D. J. Verschuur, and L. Eisner, "Machine learning in microseismic monitoring," *Earth-Science Reviews*, vol. 239, p. 104371, 2023.

[33] J. Jiang, V. Stankovic, L. Stankovic, E. Parastatidis, and S. Pytharouli, "Microseismic event classification with time, frequency and wavelet domain convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–14, Apr. 2023.

[34] S. M. Mousavi, Y. Sheng, W. Zhu, and G. C. Beroza, "Stanford earthquake dataset (stead): A global data set of seismic signals for ai," *IEEE Access*, 2019.

[35] W. Wang, P. Chen, Y. Xu, and Z. He, "Active-mtsad: Multivariate time series anomaly detection with active learning," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2022, pp. 263–274.

[36] Z. Li, Y. Zhao, Y. Geng, Z. Zhao, H. Wang, W. Chen, H. Jiang, A. Vaidya, L. Su, and D. Pei, "Situation-aware multivariate time series anomaly detection through active learning and contrast vae-based models in large distributed systems," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2746–2765, 2022.

[37] Q. Gu, Q. Dai, H. Yu, and R. Ye, "Integrating multi-source transfer learning, active learning and metric learning paradigms for time series prediction," *Applied Soft Computing*, vol. 109, p. 107583, 2021.

[38] S. Jarl, L. Aronsson, S. Rahrovani, and M. H. Chehreghani, "Active learning of driving scenario trajectories," *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104972, 2022.

[39] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.

[40] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

[41] N. Batra, R. Kukunuri, A. Pandey, R. Malakar, R. Kumar, O. Krystalakos, M. Zhong, P. Meira, and O. Parson, "Towards reproducible state-of-the-art energy disaggregation," in *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*, 2019, pp. 193–202.

[42] A. Reinhardt and C. Klemenjak, "How does load disaggregation performance depend on data characteristics? insights from a benchmarking study," in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, 2020, pp. 167–177.

[43] J. Jiang, Q. Kong, M. D. Plumbley, N. Gilbert, M. Hoogendoorn, and D. M. Roijers, "Deep learning-based energy disaggregation and on/off detection of household appliances," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 3, pp. 1–21, 2021.

[44] D. Murray, L. Stankovic, V. Stankovic, S. Lulic, and S. Sladojevic, "Transferability of neural network approaches for low-rate energy disaggregation," in *ICASSP 2019-2019 IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 8330–8334.

[45] Y. Pan, K. Liu, Z. Shen, X. Cai, and Z. Jia, "Sequence-to-subsequence learning with conditional gan for power disaggregation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3202–3206.

[46] K. Chen, Y. Zhang, Q. Wang, J. Hu, H. Fan, and J. He, "Scale- and context-aware convolutional non-intrusive load monitoring," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2362–2373, 2019.

[47] L. Massidda, M. Marrocu, and S. Manca, "Non-intrusive load disaggregation by convolutional neural network and multilabel classification," *Applied Sciences*, vol. 10, no. 4, p. 1454, 2020.

[48] İ. H. Çavdar and V. Faryad, "New design of a supervised energy disaggregation model based on the deep neural network for a smart grid," *Energies*, vol. 12, no. 7, p. 1217, 2019.

[49] Z. Yue, C. R. Witzig, D. Jorde, and H.-A. Jacobsen, "Bert4nilm: A bidirectional transformer model for non-intrusive load monitoring," in *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*, 2020, pp. 89–93.

[50] L. Wang, S. Mao, and R. M. Nelms, "Transformer for non-intrusive load monitoring: Complexity reduction and transferability," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 987–18 997, 2022.

[51] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, vol. 25, no. Citeseer. Citeseer, 2011, pp. 59–62.

[52] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Scientific data*, vol. 4, no. 1, pp. 1–12, 2017.

[53] J. Kelly and W. Knottenbelt, "The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.

[54] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware energy disaggregation using adaptive bidirectional lstm models," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3054–3067, 2020.

[55] L. Wang, S. Mao, B. M. Wilamowski, and R. M. Nelms, "Pre-trained models for non-intrusive appliance load monitoring," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 1, pp. 56–68, 2021.

[56] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1419–1429, 2020.

[57] X. Jin, "Active learning framework for non-intrusive load monitoring," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2016.

[58] A. Filip *et al.*, "Blued: A fully labeled public dataset for event-based nonintrusive load monitoring research," in *2nd workshop on data mining applications in sustainability (SustKDD)*, vol. 2012, 2011.

[59] F. Liebgott and B. Yang, "Active learning with cross-dataset validation in event-based non-intrusive load monitoring," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 296–300.

[60] A. M. Fatouh, O. A. Nasr, and M. Eissa, "New semi-supervised and active learning combination technique for non-intrusive load monitoring," in *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*. IEEE, 2018, pp. 181–185.

[61] L. Guo, S. Wang, H. Chen, and Q. Shi, "A load identification method based on active deep learning and discrete wavelet transform," *IEEE Access*, vol. 8, pp. 113 932–113 942, 2020.

[62] J. Gao, S. Giri, E. C. Kara, and M. Bergés, "Plaid: a public dataset of high-resoultion electrical appliance measurements for load identification research: demo abstract," in *proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, 2014, pp. 198–199.

[63] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen, "Whited-a worldwide household and industry transient energy data set," in *3rd International Workshop on Non-Intrusive Load Monitoring*, 2016, pp. 1–4.

[64] T. Picon, M. N. Meziane, P. Ravier, G. Lamarque, C. Novello, J.-C. L. Bunetel, and Y. Raingeaud, "Cooll: Controlled on/off loads library, a public dataset of high-sampled electrical signals for appliance identification," *arXiv preprint arXiv:1611.05803*, 2016.

[65] L. Fabri, D. Leuthe, L.-M. Schneider, and S. Wenninger, "Fostering non-intrusive load monitoring for smart energy management in industrial applications: an active machine learning approach," *Energy Informatics*, vol. 8, no. 1, pp. 1–26, 2025.

[66] J. Li, L. Stankovic, S. Pytharouli, and V. Stankovic, "Automated platform for microseismic signal analysis: Denoising, detection, and classification in slope stability studies," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7996–8006, 2020.

[67] W.-Y. Liao, E.-J. Lee, C.-C. Wang, P. Chen, F. Provost, C. Hibert, J.-P. Malet, C.-R. Chu, and G.-W. Lin, "Rocknet: Rockfall and earthquake detection and association via multitask learning and transfer learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023.

[68] X. Gu, W. Lu, Y. Ao, Y. Li, and C. Song, "Seismic stratigraphic interpretation based on deep active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023.

[69] G. F. Manley, T. A. Mather, D. M. Pyle, D. A. Clifton, M. Rodgers, G. Thompson, and J. M. Londono, "A deep active learning approach to the automatic classification of volcano-seismic events," *Frontiers in Earth Science*, vol. 10, p. 807926, 2022.

[70] S. Sykiotis, M. Kaselimi, A. Doulamis, and N. Doulamis, "Electricity: An efficient transformer for non-intrusive load monitoring," *Sensors*, vol. 22, no. 8, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/8/2926

[71] "Smart Metering Equipment Technical Specifications Version 2," 2013, https://www.gov.uk/government/publications/smart-metering-implementation-programme-information-leaflet, (Accessed on 19 June 2023).

[72] K. Pankow, "Utah forge: 2022 well stimulation seismicity data including segy data," Geothermal Data Repository, University of Utah Seismograph Stations, https://gdr.openei.org/submissions/1494, 2023, (Accessed on 29 April 2025). [Online]. Available: https://gdr.openei.org/submissions/1494

[73] P. Shi, F. Grigoli, F. Lanza, G. C. Beroza, L. Scarabello, and S. Wiemer, "Malmi: An automated earthquake detection and location workflow based on machine learning and waveform migration," *Seismological Society of America*, vol. 93, no. 5, pp. 2467–2483, 2022.

[74] S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, "Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking," *Nature communications*, vol. 11, no. 1, p. 3952, 2020.

[75] French Landslide Observatory – Seismological Datacenter / RESIF, "Observatoire multi-disciplinaire des instabilités de versants (omiv)," 2006. [Online]. Available: https://seismology.resif.fr/networks/#/MT

[76] F. Provost, C. Hibert, and J.-P. Malet, "Automatic classification of endogenous landslide seismicity using the random forest supervised classifier," *Geophysical Research Letters*, vol. 44, no. 1, pp. 113–120, 2017.

[77] P. P. M. do Nascimento, "Applications of deep learning techniques on nilm," *Diss. Universidade Federal do Rio de Janeiro*, 2016.

[78] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[79] H. Rafiq, X. Shi, H. Zhang, H. Li, and M. K. Ochani, "A deep recurrent neural network for non-intrusive load monitoring based on multi-

feature input space and post-processing," *Energies*, vol. 13, no. 9, p. 2195, 2020.

[80] D. Li, J. Li, X. Zeng, V. Stankovic, L. Stankovic, C. Xiao, and Q. Shi, "Transfer learning for multi-objective non-intrusive load monitoring in smart building," *Applied Energy*, vol. 329, p. 120223, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261922014805

[81] H. Li, Y. Wang, Y. Li, G. Xiao, P. Hu, and R. Zhao, "Batch mode active learning via adaptive criteria weights," *Applied Intelligence*, vol. 51, pp. 3475–3489, 2021.

[82] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Int. Conf. Machine Learning*. PMLR, 2016, pp. 1050–1059.

[83] C.-A. Brust, C. Käding, and J. Denzler, "Active and incremental learning with weak supervision," *KI-Künstliche Intelligenz*, vol. 34, pp. 165–180, 2020.

[84] J. Mukhoti, A. Kirsch, J. Van Amersfoort, P. H. Torr, and Y. Gal, "Deep deterministic uncertainty: A new simple baseline," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 384–24 394.

[85] G. Tanoni, E. Principi, and S. Squartini, "Multilabel appliance classification with weakly labeled data for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 14, no. 1, pp. 440–452, 2023.

[86] Y. Yang, J. Zhong, W. Li, T. A. Gulliver, and S. Li, "Semisupervised multilabel deep learning based nonintrusive load monitoring in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 6892–6902, 2019.

[87] C. Klemenjak, S. Makonin, and W. Elmenreich, "Towards comparability in non-intrusive load monitoring: On data and performance evaluation," in *2020 IEEE power & energy society innovative smart grid technologies conference (ISGT)*. IEEE, 2020, pp. 1–5.

[88] B. Ghai, Q. V. Liao, Y. Zhang, R. Bellamy, and K. Mueller, "Explainable active learning (xal): Toward ai explanations as interfaces for machine teachers," *Proc. ACM Hum.-Comput. Interact.*, vol. 4, no. CSCW3, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3432934

[89] "Delivering the European Green Deal," Aug 2022, https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal/delivering-european-green-deal_en (Accessed on 1 November 2022).

[90] A. Tonnellier, A. Helmstetter, J.-P. Malet, J. Schmittbuhl, A. Corsini, and M. Joswig, "Seismic monitoring of soft-rock landslides: the super-sauze and valoria case studies," *Geophysical Journal International*, vol. 193, no. 3, pp. 1515–1536, 2013.

[91] J. Gomberg, W. Schulz, P. Bodin, and J. Kean, "Seismic and geodetic signatures of fault slip at the slumgullion landslide natural laboratory," *Journal of Geophysical Research: Solid Earth*, vol. 116, no. B9, 2011.

[92] D. Murray, L. Stankovic, V. Stankovic, S. Pytharouli, A. White, B. Dashwood, and J. Chambers, "Characterisation of precursory seismic activity towards early warning of landslides via semi-supervised learning," *Scientific Reports*, vol. 15, no. 1, p. 1026, 2025.

[93] T. Sobot, D. Murray, V. Stankovic, L. Stankovic, and P. Shi, "An active learning framework for microseismic event detection," in *2024 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2024.

[94] L. Trani, G. A. Pagani, J. P. P. Zanetti, C. Chapeland, and L. Evers, "Deepquake—an application of cnn for seismo-acoustic event classification in the netherlands," *Computers & Geosciences*, vol. 159, p. 104980, 2022.

[95] D. Murray, L. Stankovic, and V. Stankovic, "Supervised microseismic event detection using siamese networks for labelling of noisy recordings," University of Strathclyde, WorkingPaper, Mar. 2024.

[96] Y. Li, C. L. P. Chen, and T. Zhang, "A survey on siamese network: Methodologies, applications, and opportunities," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 994–1014, 2022.

[97] W. Liu, B. Dai, A. Humayun, C. Tay, C. Yu, L. B. Smith, J. M. Rehg, and L. Song, "Iterative machine teaching," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2149–2158.

[98] J. Jiang, "Seismic signal classification and detection based on deep learning," Ph.D. dissertation, University of Strathclyde, 2025.

[99] T. Perol, M. Gharbi, and M. Denolle, "Convolutional neural network for earthquake detection and location," *Science Advances*, vol. 4, no. 2, p. e1700578, 2018.

[100] G. Tanoni, "Deep learning techniques for edge-centric non-intrusive load monitoring," Ph.D. dissertation, Universita Politecnica delle Marche, 2024. [Online]. Available: https://hdl.handle.net/20.500.14242/165896