A proposed real-time automatic in-situ acoustic anomaly detection method for the condition monitoring of remote vertical turbine pump stations

Silas Curfman, *Masters Student, The University of New Mexico,*

yy

***Abstract*—The condition monitoring of vertical turbine pump stations, found on large irrigation systems in remote areas presents a number of challenges that make it well suited for emerging techniques in the fields of machine learning, embedded systems, and machine condition monitoring. Such pump stations are often remotely located, remotely controlled, serve mission critical roles to both private and public interests, and are very costly to repair in both time and labor. In addition to the cost of unplanned repairs, small degredations in pump efficiency accumulate over time and can be difficult to notice over short term observations. Unchecked, such variations represent tens of thousands of dollars in avoidable losses.**

Such pump systems would greatly benefit from modern machine condition monitoring methods. The fact that these systems are often installed in remote areas, exposed to the elements, have little if any network connectivity, as well as the shear size of such pumps make the application of existing machine condition monitoring methods very difficult.

Proposed here is an automatic system that can be installed near, yet not interfere with, such pump stations, outside of the control loop, which can monitor for and report on multicple classes of nominal operation as well as detect anomalous running conditions that may indicate progressive failure.

***Index Terms*—Acoustic emission, Anomaly detection, Machine learning, Condition monitoring, Real-time, Signal Processing, Spectrogram, Acoustic signal processing, Embedded systems,**

I. Introduction

A. Vertical Turbine Pumps

Because of their efficency, almost all large scale irrigation systems rely on vertical turbine pump technology. A vertical turbine pump is a type of centrifugal pump built to be submerged in water [17], are driven by direct mounted electric motors, and capable of lifting large volumes of water at greater efficiencies than traditional centrifugal pumps. Vertical turbine pumps can be found in many different sizes and applications.

A single Vertical turbine pump consists of the following elements, typically found from bottom (input) to top (output): a suction filter or screen, a series of pump stages called the bowl assembly, the column assembly, and finally the head assembly. Atop the pump assembly, a large electric motor is coupled to a drive shaft that extends the full length of the column and bowl assemblies; this shaft drives the impellers housed inside the bowls of the bowl assemby. Attached to the drive motor is typically a variable frequency drive, which in turn is networked to some form control system or programmable logic controller. In addition to controlling the operation of the pump, the control system is often able to report on operating parameters such as drive frequency, voltage, amperage, and motor rpm. Such pumps and control systems can be run individually, but are more often installed as modules within a large array of such pumps.

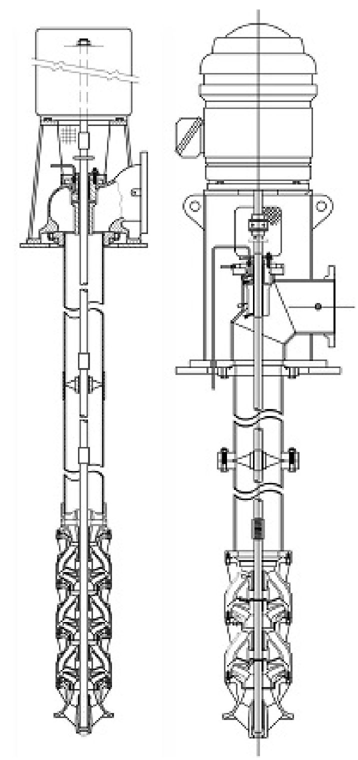

Fig. 1: Vertical Turbine Pump Cross Section

B. Vertical Turbine Pump Stations

A vertical turbine pump station represents a system in which multiple vertical turbine pumps are linked together. These stations draw input water from the same source, and combine the outputs into a common manifold or reservoir depending on design. During operation, depending on demand, the number of pumps engaged may range from all, to some, or even just one. As crop growing cycles are inflexibly linked to annual cycles, and absence of water can permanently alter soil conditions not just reduce crop growth; up-time availability of these systems is considered extremely critical. Standard operating procedures involve predictive maintenance and rebuilds of individual pumps during non-growing seasons.

Year to year, the operators of these systems will select a number of pumps to be removed and sent to machine shops that specialize in the tear down, inspection, repair, re-assembly, testing, and installation of such pumps. The down time for a full rebuild is typically measured in months. Even when scheduled in advance, the rebuild process is extremely costly and often disruptive to nominal operating schedules.

A system, if designed to continuously monitor such pump stations, able to classify observed running conditions according to known nominal classes, capable of identifying non-nominal operating conditions absent any a priori knowledge, and able carry out these duties automatically while under harsh conditions in remote regions; would have a very high return on investment in terms of machine health monitoring and large scale predictive maintenance.

Fig. 2: Multi-Unit Pumping Station

C. The Problem

The problem with designing such a system up to this point has been a combination of challenges. The remote nature of these installations has traditionally meant extremely limited compute ability. Both on-premise and cloud based solutions available to facility based installations are typicall not an option for such systems. Further challenging is limited, meaningful, operating parameters which can be safely monitored or recorded without having to either interfere with the operation of the pumps or share bandwidth with the critical control system or PLC network. Any condition monitoring parameters need to be observable from at least some minimum distance yet also provide useful insight into the machine health. Finally, while there are some general consistencies from one such installation to the next, there are far more variations between them. Therefore in order to scale accordingly, any classification rubric would have to demonstrate a degree of flexibility or tunability that could accomodate difference in baseline operation observations from installation to installation. In short, any classification algorithm would have to allow for customization to different pump stations locales, configurations, and environmental operating conditions. The aim of this work is to study the feasibility of one such proposed design solution. The proposed solution will use vertical pump stations as a hypothetical use case and focus on using acoustic signals in the human perceivable range as inputs to a flexible machine learning implementation that can ultimately be scaled down to a size suitable for a low power embedded micro-controller.

II. Background

A. Machine Condition Monitoring

The cross domain nature of this project easily divides along the following sub domains; Machine Condition Monitoring, Machine Learning, and Embedded Systems (IoT). A literature review was done within each domain in order to ascertain what types of methods and resources have most recently been put to use in similar projects.

TABLE I: Literature Domain Matrix.

Reference	MCM	ML	IoT
muller2020acoustic	✗	❍	❍
potovcnik2021condition	✗	❍	✗
dohi2022description	✗	❍	✗
janssens2017deep	✗	❍	❍
ince2010machine	✗	❍	✗
thomas2019hierarchical	✗	✗	❍
ren2021tinyol	✗	✗	❍
karges2022soundscapes	❍	✗	❍
tsuji2021machine	❍	✗	✗
tagawa2021acoustic	✗	❍	✗
mulimani2018acoustic	✗	❍	✗
purohit2019mimii	✗	❍	✗
ray2021review	✗	✗	❍
murshed2021machine	✗	✗	❍
barksdale2018condition	❍	✗	✗
...	...	...	...

Literature on machine condition monitoring produced a large body of research on the subject. The initial results were then reduced to literature focusing on acoustic methods in particular Further filtering was done to categorize the literature as either (i) including machine learning methods [3][12][19] or (ii) including embedded systems deployments [21].

[14],[8],[12] all applied audio to image transformations on the raw input data. Their image transformations performed well when used in the same type of neural networks that had previously shown good performance on natively digital images. Their methods varied most in the digital signal processing portion. During the signal processing phase, a number of image representations were tested. These include Spectrograms, Mel-Frequency Spectrograms (MFE), Mel-Frequency Cepstral (MFCC) Coefficients, and more. The most effective tended to be split between MFE and MFCC representations. On embedded systems [6] and [2] demonstrated a number of practical implementations of machine learning applied to machine condition monitoring at a scale which could be deployed to small embedded systems.

Also, [10] presented a high level survey of a number of proven implementations and their associated workflows. Unfortunately very little of the literature reviewed applied to acoustic methods and even less so to the psychoacoustic range.

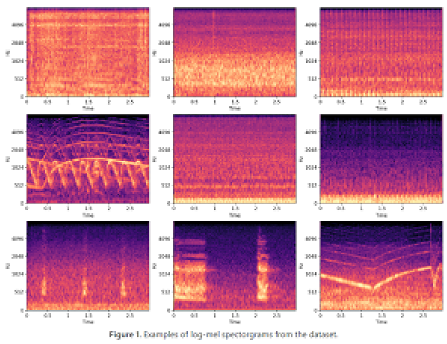

Fig. 3: MEL-Frequency Spectrograms

B. Machine Learning Methods

Literature on machine learning methods applied to acoustic signals in general was also reviewed This result was then further refined to papers including acoustic data in the human discernable (psychoacoustic) range. These search were then categorized as either (i) multi-label classification problems such as, (ii) anomaly detection problems, or both.

[13], [5] and [4] all demonstrated valid implementations of audio classification. [11], [18], and [1] all demonstrated working applications of audio anomaly detection. Literature involving both classification and anomaly detection of audio inputs was very limited.

C. Embedded Systems / IoT

A rapidly growing, yet still nascent, body of literature is being produced on the topic of applying machine learning methods to low power embedded systems. [21] tested the relative performance of different wavelets used in the signal processing work flows in search of high performance with low power cost overhead. Both [20] and [16] demonstrate practical workflows for training, testing, optimizing deep learning algorithms in the cloud and then inferring the optimized models down to the edge on to low power micro-controller devices.

The literature review of these three domains shows both fairly well established methods and approaches; yet provide very few examples that could be applied to apply to all three. There is relatively little literature available on the use of machine learning on raw audio as the input. Likewise, of the embedded systems literature, very little of it touches on machine learning. At this time, literature on acoustic based anomaly detection deployed on embedded systems seems to be the most scarce in particular.

From the review of the current literature, it is clear that there is a need to develop end to end workflows that can be applied to audio classification and anomaly detection in the psychoacoustic range. In this context, this paper explores one such possible workflow from end to end. The details of the experiment, its setup, methods, and materials follow in the next section.

III. Methods and Materials

To solve the challenges described above, the method presented attempts to balance a number of trade-offs between these three domains. The first compromise was to specify a hardware component that both required very little in terms of power, compute, and network resources; while also providing a suitable environment to run a machine learning classification and anomaly detection algorithm on. The networking and computational limitations make this an excellent application for a micro-controller (MCU) based system capable of running a highly optimized (i.e. TinyML), pre-trained, machine learning algorithm. By separating the training and testing from the live classification tasks, much higher computational resources such as GPU's can be leveraged to build the initial model while retaining the low power advantage of the MCU for deployment. In this case the Arduino BLE 33 Sense programmable circuit board [Fig. 4] was an excellent choice. Not only does the Arduino Nano BLE 33 Sense provide a low power MCU capapble of hosting machine learning tasks through the use of TensorFlow Lite, the Nano BLE 33 Sense also comes pre-installed with a number of embedded sensors such as a 9-axis inertial sensor and a MEMS microphone.

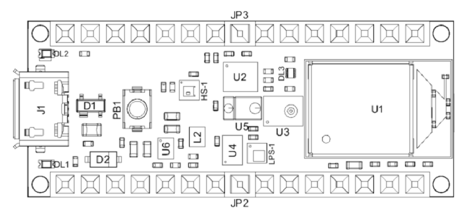

Fig. 4: Arduino Nano BLE 33 Sense

Only after the hardware environment had been selected could the field of Machine Condition Monitoring (MCM) methods be narrowed down to a selection practical for this project. With the presence of the embedded 9-axis inertial sensor and the MEMS microphone, two obvious options for MCM methods were vibration analysis and acoustic analysis. While the 9-axis IMU sensor would be a great choice, the sensor would need to be installed directly on the pumps by some means. The need to keep any installed sensors from interfering with the pumps in any way, including direct physical access, makes the IMU sensor less attractive. Fortunately the embedded MEMS microphone provides a way to do acoustic analysis while maintaining a minimum physical distance from the actual pumps. Acoustic based MCM methods typically distinguish along the lines of frequency ranges. From infrasound (less than 20 Hz) at the low frequency end to ultrasound (higher than 20 KHz) at the high frequency end, a number of different approaches exist. Because ultrasound requires very close installation of the sensors, and because infrasound is too low a range to apply to the rotating machinery of these pumps, the psychoacoustic range was selected. The psychoacoustic range, from 20 Hz - 20 KHz, also maps directly to human perceivable sounds [Fig. 5]. This unique characteristic can be leveraged during data set collection and the sample labeling process. At this frequency range we can take advantage of a human-in-the-loop approach and utilize the expertise of the operators and maintainers of these installations to help build accurately labeled data sets.

Finally, having defined the computational resources avail-

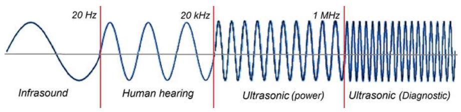

Fig. 5: Sound Spectrum

able and also selected the physical phenomena that we will be using for our data sets, the large number of machine learning methods can be narrowed down to a handful of approaches that could fit our build. The pairing of deep learning methods with low power micro-controllers, also known as TinyML [15] and [16] provides an excellent starting point for algorithm selection. Especially useful is the work developed in speech recognition through which it has been shown that if an audio signal can be transformed into an image representation, then the same deep learning methods used for visual classification can be applied to acoustic classification problems [9] [7] [18] [22].

A review of the literature shows that the likely best choices to start with involve converting the audio signal to a Mel-Frequency Engery Spectrogram (MFE) or Mel-Frequency Cepstral Coefficients image representation. This process is repeated at specific time intervals, advance at a specified rate, scan the entirety of the sound sample, and produce a number of 2 dimensional RGB images of a fixed pixel by pixel resolution. According to [22]

> ... two-dimensional stack of MFCC representations enable the capturing of time-frequency characteristics of the input signal

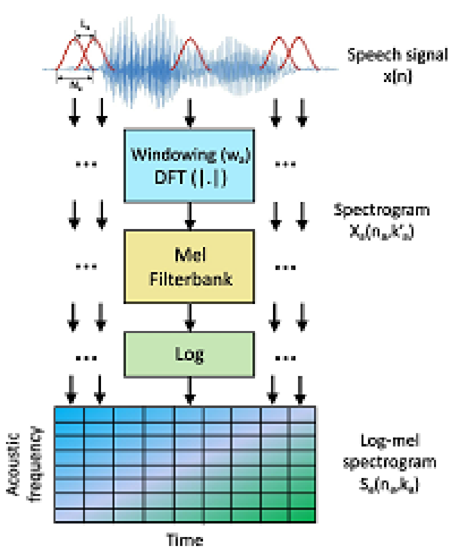

Fig. 6: Audio to Image

Every sound sample of every label produces a multitude of images representing the same class label as the parent recording. These images are then used as training and testing inputs for a deep learning convolutional network [Fig. 6][Fig. 7]. This workflow defines the classification process.

Extending the machine learning to anomaly detection has been shown to be more challenging than strict classification if also using an image representation approach such as MFE or MFCC. This is because the image representation process creates a very large number of feature parameters. This high volume of feature parameters makes established anomaly detection methods such as K-Means, Support Vector Machines (SVM), and Local Outlier Factor (LOF) difficult to apply. Some success has been shown though by feature extraction of a 2d spectral analysis wave plot rather than an RGB image created via MFE or MFCC. A spectral analysis approach was used in conjunction with a K-Means classifier here for anomaly detection.

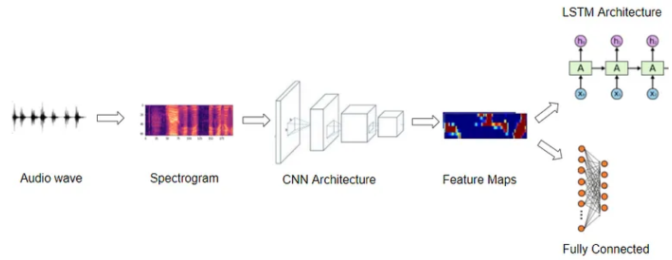

Fig. 7: Acoustic Feature Extraction

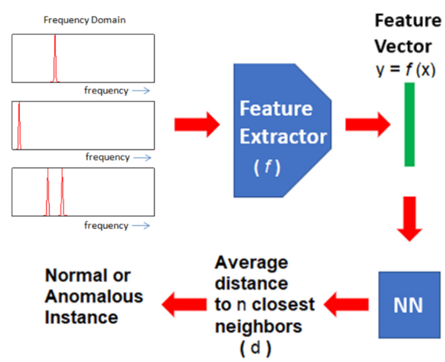

Fig. 8: Spectral Feature Extraction

IV. Results

A. Dataset

For the purposes of this project a dataset was built from scratch. While using a pre-existing dataset such as MIMII [14] would be much less work. One of the defining challenges of this project though, is the relative variation between pump stations and their installation environments. As such, the choice was made to include all the steps necessary to collect a dataset unique to a given installation.

To simulate the acoustics of submersible pump machinery without routine access to production units, a miniature system was put together using a decorative fountain and electric submersible pump. The fountain pump was variable speed, had a built in debris screen, and was easily positioned in different arrangements. An external USB microphone was mounted on a tripod 5-10 inches away from the pump and fountain for initial data collection. By reconfiguring the pump and fountain it was possible to simulate 4 nominal running classes and 1 anomaly running class. The nominal running classes were: Idle (i), Speed 1 (s1), Speed 2 (s2), and Dry Running (dr). A fifth class called Cavitation (cav) was recorded and held out of the nominal group. The 4 nominal classes were used for training the deep learning classifier through the use of image representations via MFE. The fifth class was used to train the anomaly detection algorithm using spectral analysis as an input to a K-means classifier [Table II].

The audio data collection was done in 4 minute blocks with one block per classification. These were saved in a collection labeled 'raw' and named individually according to their class. The 4 minute master sets were then broken down into 24 10 second clips each. These clips were then reviewed manually for any recording errors. A small number of clips were rejected due to minor recording errors. The remaining 10 second clips were then divided into a roughly 75% : 25% testing : training split.

TABLE II: Dataset Description

Set	Num	m:s	Freq	Labels
RAW	5	4:00	44 KHz	i s1 s2 dr cav
TRAIN	56	0:10	44 KHz	i, s1, s2, dr, cav
TEST	19	0:10	44 KHz	i, s1, s2, dr, cav
ANOMALY	21	0:10	44 KHz	cav

B. Results

1) Classification

a) Signal Processing: The primary objective when setting the signal processing parameters was to arrive at a solution that both provided the deep learning algorithm with image frames that had a high visual distinction between classes while being very similar within classes. Additionally, the signal processing parameter needs to generate as few feature parameters as possible in order to keep the inference time of the deep learning algorithm to a reasonalbe overhead for the Coretex M0+ processor of the Arduino. The final settings for the signal process phase can be seen in the last row of [Table III].

TABLE III: Signal Preprocessing

Iteration Filter Num.	Time Frame FFT Length	Time Stride Low Band	FFT Frame High Band	FFT Stride Features
Attempt 1	2000ms	1000ms	0.05	0.05
32	256	300 Hz	0 Hz	16,000
Attempt 2	2000ms	2000ms	0.075	0.075
NA	128	NA	NA	88,200
Attempt 3	1000ms	500ms	0.05	0.05
32	256	300 Hz	0 Hz	16,000

b) CNN Structure: The objective when chosing a deep learning algorithm and structure was to balance precision with the performace limitations of the Coretex M0+. A number of iterations of different structures were experimented with. A selection of high precision structures were then attempted to flash to the Arduino. Ultimately a small set of 2D Convolutional Neural Networks programmed in TensoFlow Lite were successfully flashed to the Arduino and were able to run correctly. Of these few, the TensoFlow Lite model that consumed the least run time resources on the Arduino was chosen as the best canditate.
The final deep learning model was a 2D CNN with (1) reshape layer, (2) convolutional layers, (1) flattening layer, and an output layer of (4) classes. A sample runtime output of this model can be seen in [Fig. 9]

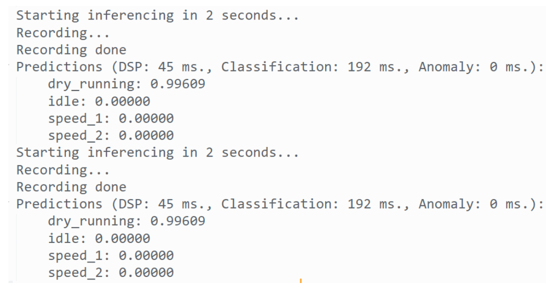

Fig. 9: Serial Terminal Output

c) Classification Performance: While training the CNN the off chip simulation results were consistently high with a confusion matrix showing an expected accuracy of 100% across all four classes. This was clearly optomistic as we could only simulate what the classifier may run like once flashed. As expected, once flashed to the Arduino and placed in a live sound capture demo, the performance showed a number of misclassifications. A comparison between the simulation performance using pre-recorded samples and the deployed demo using live audio can be seen below.

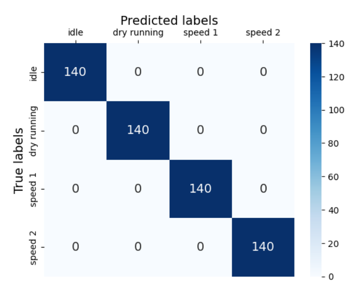

Fig. 10: Simulated Precision

2) Anomaly Detection (TBD)
At the time of this printing, the anomaly detection algorithm had not been refined to a model capable of being scaled down small enough for the Coretex M0+ processor on the Arduino.

a) Signal Processing: It was shown that while still generating features in the hundreds rather than 10's of thousands; by using a spectral analysis plot as the input to a K-Means classifier a handful of generated features could be selected as tuning

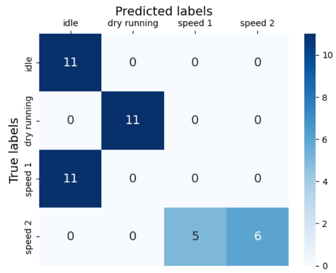

Fig. 11: Live Demo Precision

parameters. After doing a few simulations, a set of possible combinations was being narrowed down to. One such simulation is shown below. There remains more work to do on the anomaly detection algorithm but indications are that it can be achieved with enough trial and error trading of precision and performance.

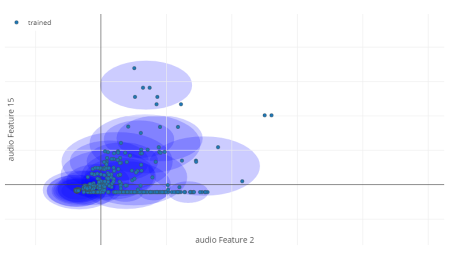

Fig. 12: K-means Simulation

b) K-Means Structure: TBD
c) Anomaly Detection Performance: TBD

V. Discussion

1) While success was made by pairing off device deep learning with very small onboard inference models for classification, doing the same with anomaly detection has proved very challenging. The image based CNN structure had no difficultly with the large number of parameters generated by the audio to image pre-processing, but such a high number of parameters also invokes the curse of dimensionality when trying to build a one-class classifier for anomaly detection. The scope of this project cover only K-means as an anomaly detection algorithm, future work could certainly be done both more deeply on K-means but also by incorporating other algorithms such as LOF, SVM, or possibly LSTM.
2) The results in this paper are heavily influenced by the proposed use case stated in the introduction. While the application of this method to vertical turbine pump stations may seem very niche, such pump stations are simply one example of large electric driven rotating machinery found throughout every industry. This workflow should be transferable to any piece of machinery that continuously runs in some predictable nominal states and produces enough operating noise to be picked up by a properly placed and filtered microphone.

   The methods described here would not work well for discrete acoustic classification and anomaly detection. Sudden transient acoustic events such as a car crash, automotive horn, or gun shots would not be cases where these methods would apply.

VI. Conclusion

In this work we took an inquiry about a potential use case for acoustical monitoring machinery condition on systems installed in remote locations with limited computation and networking resourses, built a design specification around the use case, designed and built a prototype, and conducted deployed experiments testing a number of different methods and algorithms.

The use of TinyML methods for pairing resource intensive deep learning methods with low power microcontrollers proved to be extremely useful. The TinyML domain is still only a few years old and rapidly growing. It is very probable that in the next year, if not already now, a new approach to anomaly detection using sound would be developed that could be directly incorporated into this project.

Future work should also be done in dataset creation of acoustic machine condition monitoring cases. The process of such dataset collection could and should be distributed to the public, and leverage the domain expertise of the owners and operators of large industrial rotating machinery.

references.bib

References

[1] Francesco Aurino et al. "One-class SVM based approach for detecting anomalous audio events". In: *2014 International Conference on Intelligent Networking and Collaborative Systems*. IEEE. 2014, pp. 145–151.

[2] Hunter Barksdale, Quinton Smith, and Muhammad Khan. "Condition monitoring of electrical machines with Internet of Things". In: *SoutheastCon 2018*. IEEE. 2018, pp. 1–4.

[3] Kota Dohi et al. "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques". In: *arXiv preprint arXiv:2206.05876* (2022).

[4] Nuri Firat Ince et al. "A machine learning approach for locating acoustic emission". In: *EURASIP Journal on Advances in Signal Processing* 2010 (2010), pp. 1–14.

[5] Olivier Janssens et al. "Deep learning for infrared thermal image based machine health monitoring". In: *IEEE/ASME Transactions on Mechatronics* 23.1 (2017), pp. 151–159.

[6] Nils Karges et al. "Soundscapes on edge-The real-time machine learning approach for measuring Soundscapes on resource-constrained devices". In: *PROCEEDINGS of the 24th International Congress on Acoustics*. 2022, pp. 128–139.

[7] Yuma Koizumi et al. "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection". In: *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE. 2019, pp. 313–317.

[8] Manjunath Mulimani and Shashidhar G Koolagudi. "Acoustic event classification using spectrogram features". In: *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE. 2018, pp. 1460–1464.

[9] Robert Müller et al. "Acoustic anomaly detection for machine sounds based on image transfer learning". In: *arXiv preprint arXiv:2006.03429* (2020).

[10] MG Sarwar Murshed et al. "Machine learning at the network edge: A survey". In: *ACM Computing Surveys (CSUR)* 54.8 (2021), pp. 1–37.

[11] Eduardo C Nunes. "Anomalous sound detection with machine learning: A systematic review". In: *arXiv preprint arXiv:2102.07820* (2021).

[12] Rodrigo de Paula Monteiro et al. "A hybrid prototype selection-based deep learning approach for anomaly detection in industrial machines". In: *Expert Systems with Applications* 204 (2022), p. 117528.

[13] Primož Potočnik et al. "Condition classification of heating systems valves based on acoustic features and machine learning". In: *Applied Acoustics* 174 (2021), p. 107736.

[14] Harsh Purohit et al. "MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection". In: *arXiv preprint arXiv:1909.09347* (2019).

[15] Partha Pratim Ray. "A review on TinyML: State-of-the-art and prospects". In: *Journal of King Saud University-Computer and Information Sciences* (2021).

[16] Haoyu Ren, Darko Anicic, and Thomas A Runkler. "Tinyol: Tinyml with online-learning on microcontrollers". In: *2021 international joint conference on neural networks (IJCNN)*. IEEE. 2021, pp. 1–8.

[17] Thomas F Scherer. "Irrigation water pumps". In: (1993).

[18] Kaori Suefusa et al. "Anomalous sound detection based on interpolation deep neural network". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 271–275.

[19] Yuki Tagawa, Rytis Maskeliūnas, and Robertas Damaševičius. "Acoustic anomaly detection of mechanical failures in noisy real-life factory environments". In: *Electronics* 10.19 (2021), p. 2329.

[20] Anthony Thomas et al. "Hierarchical and distributed machine learning inference beyond the edge". In: *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*. IEEE. 2019, pp. 18–23.

[21] Kazuki Tsuji et al. "A machine sound monitoring for predictive maintenance focusing on very low frequency band". In: *SICE Journal of Control, Measurement, and System Integration* 14.1 (2021), pp. 27–38.

[22] Alexander Wong et al. "Tinyspeech: Attention condensers for deep speech recognition neural networks on edge devices". In: *arXiv preprint arXiv:2008.04245* (2020).