Biomedical Signal Processing and Control 105 (2025) 107556

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Autonomous collection of voiding events for sound uroflowmetries with machine learning

Laura Arjona [a,*], Sergio Hernández [a], Girish Narayanswamy [b], Alfonso Bahillo [c], Shwetak Patel [b]

[a] Faculty of Engineering, University of Deusto, Av. Universidades, 24, Bilbao, 48007, Spain
[b] Paul G. Allen School of Computer Science and Engineering and the Department of Electrical and Computer Engineering, University of Washington, 185 E Stevens Way NE, Seattle, 98195-2350, WA, United States
[c] Department of Signal Theory and Communications, University of Valladolid, P.° de Belén, 7, Valladolid, 47011, Spain

ARTICLE INFO

Keywords:
Acoustics
Sound sensing
IoT
Sound-based uroflowmetry
Edge computing
Machine learning

ABSTRACT

We present AutoFlow, a Raspberry Pi-based acoustic platform that uses machine learning to autonomously detect and record voiding events. Uroflowmetry, a noninvasive diagnostic test for urinary tract function. Current uroflowmetry tests are not suitable for continuous health monitoring in a nonclinical environment because they are often distressing, costly, and burdensome for the public. To address these limitations, we developed a low-cost platform easily integrated into daily home routines. Using an acoustic dataset of home bathroom sounds, we trained and evaluated five machine learning models. The Gradient Boost model on a Raspberry Pi Zero 2 W achieved 95.63% accuracy and 0.15-second inference time. AutoFlow aims to enhance personalized healthcare at home and in areas with limited specialist access.

1. Introduction

One of the problems frequently associated with ageing is that related to the urinary system. Voiding dysfunction is highly prevalent and has a major impact on the quality of life of many people (more than 60% of men over 60 years of age) [1]. Lower Urinary Track Symptoms (LUTS) are those that affect the filling and emptying of the bladder and post-voiding. They lead to a significant decrease in personal quality of life and considerable expenditure in healthcare resources. Considering that the prevalence of voiding pathologies increases with age and that the global population is ageing, it is expected that the number of males who will need medical treatment for LUTS will increase significantly in the next 20 years.

Uroflowmetry is an important screening test that can aid in the diagnosis, prognostication and follow-up of urological diseases. This test tracks how fast urine flows, how much urine flows out, and how long it takes. Current uroflowmetry tests are not suitable for continuous health monitoring in a nonclinical environment because they are often distressing, costly, and burdensome for the public. It is carried out on an outpatient basis at specified procedure areas and involves having the person urinating into an uroflowmeter. This process is unnatural and requires "on-demand" voiding, often with either low or very high bladder filling. This leads to significant test-to-test variability because the situational stress of the patient can affect the flow rate, corresponding to non-representative results [2]. Therefore, it has been recommended that the uroflowmetry test should be performed more than once, which requires time-consuming and costly repeated clinic visits. Obtaining uroflowmetry data in the home setting has the potential for increased data on voiding patterns to inform clinical decision-making [3].

This is the reason why the demand for smaller, more versatile devices has grown and led to the emergence of dedicated portable uroflowmeter. Nevertheless, these devices have not been fully adopted into routine practise, because they are costly and difficult to operate. Therefore, the envisioned platform for uroflowmetry targeted by this work should be cost-effective, easily transportable, and capable of conducting consistent tests without reliance on specialized equipment.

Recent studies have demonstrated the feasibility of using a mobile device (a smartphone [4] or a smartwatch [5]) in a home environment, to characterize the urinary flow patterns by capturing the sound generated when the urine stream hits the water in a toilet bowl. This test is known as sound uroflowmetry. The scarce literature data describing attempts to analyze sound associated with urine flow in urology has shown that sound uroflowmetry can be a viable alternative to standard uroflowmetries applying machine learning based algorithms and visual comparison, respectively [6]. However, these mobile devices need an active user interaction (users need to interact with an app), and their

* Corresponding author.
E-mail addresses: laura.arjona@deusto.es (L. Arjona), girishvn@uw.edu (G. Narayanswamy), alfonso.bahillo@uva.es (A. Bahillo).

https://doi.org/10.1016/j.bspc.2025.107556
Received 2 August 2024; Received in revised form 10 December 2024; Accepted 19 January 2025
Available online 4 February 2025

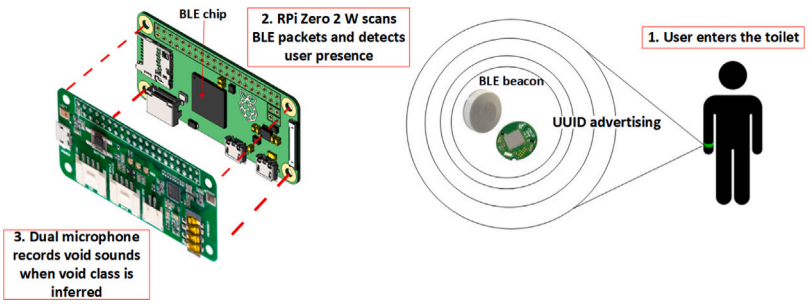

Fig. 1. Use case of the proposed platform. 1. The user wearing a BLE beacon enters the toilet. 2 The RPi detects the user presence with the BLE receiver. 3. When the users starts voiding the event is detected by the ML model running on the RPi and the recording starts.

battery needs to be recharged often, especially in the case of commercial smartwatches. Furthermore, according to [7], the adoption of mobile health (mHealth) apps in personal health care is still limited.

Overall, to detect and record voiding events automatically (without user intervention), both the smartphone and smartwatch pose important challenges:

- They need an active user interaction, requiring users to press the record button. This could create a barrier between children and the elderly. Also, this could limit data collection at night, when it is highly likely that users will forget to operate any device.
- They could create a certain user rejection due to the inconvenience of wearing or carrying the device while urinating (specially the smartphone).
- The smartphone has additional challenges such as being placed at a fixed height during recording to obtain consistent results.
- Both of them require sufficient battery power to record the event.

2. Related work

Detecting and recording voiding events automatically at home using mass market devices is a must if we want that sound uroflowmetries to support urology pathology diagnosis.

Analyzing the state of the art, it is observed that there are no works or commercial products that records void events, in a natural environment using a conventional toilet (impact sound of urination on water and not plastic), where the recording device captures audio signals in a transparent way for the user and using a low-cost device. To overcome these limitations, we have developed Auto-Flow, a proof-of-concept acoustic Raspberry Pi (RPi) based platform that runs a machine learning (ML) algorithm to automatically detect and record voiding events at home that can be easily incorporated into a normal daily routine requiring no human intervention. Furthermore, our platform can also enable the collection of events at nighttime, because users typically forget to activate any device while voiding at nighttime. This work presents two main contributions:

- ML model to detect voiding events: we predict the sound label in an audio clip and automatically collect the sound of voiding events.
- Autoflow platform: RPi-based architecture that detects the user closeness to the toilet with Bluetooth Low Energy (BLE) and run inference with the proposed ML model, engaging in edge computing by conducting model inference directly on the device.

2.1. Artificial intelligence for sound-based activity recognition

Sound-based activity recognition has gained significant traction with advancements in artificial intelligence and machine learning. These technologies enable the automatic classification and analysis of sound signals, facilitating various applications from human activity monitoring to machinery operation tracking.

The work in [8] developed an artificial neural network model to classify patterns of sound signals, enabling the recognition of human activities in indoor environments. This research highlights the potential of AI in interpreting acoustic data for activity recognition with high accuracy

A different work [9] explored sound-based crowd activity recognition using neural networks. Their research demonstrated that activities performed by humans could be recognized by the sounds emitted during these activities. This approach provides valuable insights into managing and understanding crowd dynamics through sound analysis .

Another significant area of research is the detection of anomalous sound events using machine learning techniques [10]. This includes the use of hierarchical recurrent neural networks (RNNs) to enhance audio surveillance and the application of generative adversarial networks (GANs) for anomaly detection in industrial settings. These approaches have shown effectiveness in identifying abnormal audio patterns,

3. Automatic collection of voiding events

This section presents the hardware and software platform developed for automatic void event detection and collection. All the code has been open-sourced, and currently available on [11]. Fig. 1 presents an overview of the complete platform.

3.1. Hardware platform

The designed firmware is fully RPi compatible. For continuous acoustic energy monitoring, we use the low power stereo codec ReSpeaker 2-Mics Pi HAT [12]. It is a dual-microphone expansion board for the RPi specifically designed for artificial intelligence and voice applications. The decision to use the Raspberry Pi platform in our prototype was based mostly on its widespread availability, making it an ideal choice for cost-sensitive research and development projects. Also, the RPi offers sufficient computational power to handle real-time ML inference. Additionally RPI has been used in similar IoT applications with successful results [13]. We set the sample rate to have a value of $fs = 16$ kHz, as a compromise between bandwidth, performance, and processing latency. We selected this value because similar performance was obtained at higher sampling rates, at the cost of increasing the processing latency. The RPi and the microphone are placed inside a plastic case to prevent moisture and dust to get in the boards.

To detect the presence of the person in the bathroom and the closeness to the toilet, we use a BLE beacon module that incorporates an accelerometer sensor. The user needs to wear this device all the time during the duration of the study. We selected the EMBC22 device because it is low energy and does not require maintenance from the user. According to the beacon datasheet [14], the typical current consumption in beaconing and active mode (always transmitting) is 15 μA. If we assume a CR2032 battery with a 210mAh capacity, the

battery will last around 583 days, that is, more than 1.5 years. The RPi is equipped with a BLE receiver to detect the BLE signals, allowing for proximity detection without requiring a constant connection between the beacon and the RPi. The beacon has been configured to start advertisement when movement is detected, and continue advertising BLE packets for 60 s. After that period, it waits for a new movement event to trigger advertisement again. Finally, it is important to note that the user does not need to interact with the ble beacon.

3.2. Firmware

The system, once powered, is continuously working. Once the device is powered, it automatically attempts to connect to the Internet using WiFi protected setup. If the connection is established, an LED will turn on. If the connection does not succeed or there is no WiFi network available, the platform works in offline mode. In the offline mode, the recorded audio signals are stored in the internal memory of the device, ready to be sent when an Internet connection is reestablished. Next, the program can be divided into four sequential parts.

1. User presence detection with a BLE beacon: When the RPi is powered, it scans for BLE packets for five seconds, then pauses for one second to save power. During broadcast message decoding, the program gets the received signal strength indicator (RSSI). With a constant broadcasting power value of 6.2 dB, RSSI helps approximate user proximity to the RPi using a threshold, th_{ble}. If the user's beacon address is found and $RSSI > th_{ble}$, the program starts inference to detect void or no-void sound.
2. Real-time sound classification: The model inference runs every second, using the last five seconds of buffered audio. This process continues for up to 90 s. If a void event is detected, 60 s of audio is recorded and saved as a WAV file for offline uroflowmetry analysis. If no void sound is detected in 90 s, the process stops, and the program returns to BLE scanning. Timing values can be adjusted per user.
3. Void audio transmission to the web server: If the RPi is online, detected and recorded void audio is sent to the web server and deleted from the device. If offline, audio is stored locally and uploaded once the connection is restored.
4. Sound-based uroflowmetry: The final step involves extracting the void signal envelope from the audio to determine shape and timing parameters, following the method in [5].

3.3. REST APi

The web server has been developed as a RESTful APi with the framework Flask. This framework was selected because it is very flexible and lightweight. The database has been built using MongoDB. The web server is running on an external PC machine. The source code and some screenshots are available in our repository [11].

4. Materials and methods

4.1. Dataset description

In order to classify audio events into void or no-void, we have developed a novel dataset with a total of 1420 audio signals, that is comprised of two classes: void (class 1) and no-void (class 0). Class 1 audios represents the 51.4% of the complete dataset, while class 0 audios represents the remaining 48.6%. Each audio event is a 10-second-long audio clip. Next, we detail how we built the dataset for each class.

Table 1
Audio clips of the class 0 dataset.

Audio class label	Source	# samples
Silence	This work [11]	102
Pump (liquid)	Audioset	48
Sink (filling/washing)	Audioset	48
Squish	Audioset	36
Splash, splatter	Audioset	36
Toilet Flush	Audioset	74
	ECS50	
Fill (with liquid)	Audioset	32
Pour	Audioset	71
	ECS50	
Slosh	Audioset	31
Drip	Audioset	30
Stir	Audioset	25
Spray	Audioset	24
Water tap	Audioset	23
Boiling	Audioset	16
Bathtub	Audioset	15
water drops	ESC-50	40
brushing teeth	ESC-50	40

- Class 1: it consists of 730 10-second-long voiding events audio recordings collected with the Urosound App and the Oppo smartwatch. All these audios have been collected from hospital and clinic patients as an extension of the study presented in [5]. The experimental procedures conform to the provisions of the Declaration of Helsinki (as revised in Edinburgh 2000). We split each original audio signal in 10-second-long audio clips with no overlap. This split value was selected to be consistent with Audioset [15] data (see below), used in this work for Class 0 data. We remove the first and last second of the recording to avoid discontinuities. We listened to all the clips to ensure that they correspond to audio events.
- Class 0: it consists of 690 audio events that typically occur in a traditional home toilet but which are not voiding events. To create this class dataset, we collected audios from two different open-source dataset, presented next. Table 1 shows the audios categories selected for each dataset.

    1. Audioset [15]: it consists of an expanding ontology of 632 audio event classes and a collection of 2084320 human-labeled 10-second sound clips drawn from YouTube videos. We selected the audio clips with the labels presented in Table 1.
    2. ESC-50 [16]: it is a labeled collection of 2000 five-second long environmental audio recordings. Clips in this dataset were manually extracted from public field recordings gathered by the Freesound.org project. The dataset consists of 50 classes, with 40 samples per class. We selected the audio clips with the labels presented in Table 1.
    3. Silence events: we collected 102 five-second long silence audio recordings with the Oppo smartwatch, in different toilet environments.

4.2. Sound classification model

In this section we develop a supervised learning model to automatically detect a voiding event from acoustic energy. We build a model for real time classification of incoming audio clips into two classes: void or no-void. Our model must meet three main requirements: (1)run inference in real time, (2)run on a low power device with constrained computation capabilities, and (3)be sensitive to low number of false negatives to minimize the probability of missing the collection of a voiding event The use of a low-power device with constrained computational capabilities is primarily aimed at reducing power consumption

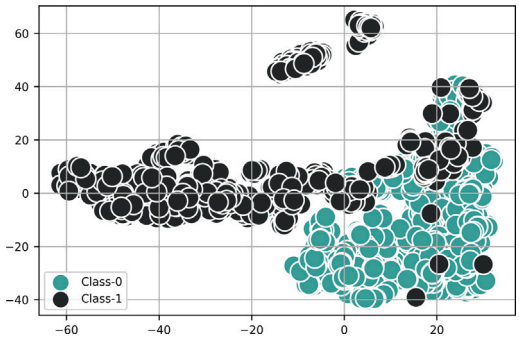

Fig. 2. t-SNE evaluation on the dataset MFCC features.

and hardware cost. Additionally, performing inference directly on the device eliminates the latencies associated with server communication.

To the best of our knowledge, there is no previous work on developing a ML model for automatic voiding event detection at home using sound as the input features and requiring no user intervention.

4.2.1. Machine learning model

Firstly, considering the size of our dataset (a total of 1420 audio samples) and the requirement to run inference in an embedded device, we explore traditional lightweight ML models for the classification task. For each audio clip we extract an array of 40 features, that correspond to 40 Mel-frequency cepstral coefficients (MFCCs) values averaged over the duration of the clip. While originally developed for speech-related tasks, MFCCs have proven highly effective for activity classification in non-speech domains, such as environmental sounds, machinery diagnostics, and biological signals. [17,18]. MFCCs are widely used audio features for training machine learning models due to their ability to effectively represent the spectral and temporal characteristics of audio signals. The MFCCs features were selected due to the compressed representation of the signal [19].

The pipeline is shown in Fig. 3 for a 10-seconds-long audio clip. To compute the MFCCs, we used the Python library librosa (function librosa.feature.mfcc) This library computes the coefficients as:

$$c_n = \sum_{k=1}^{K} \log(S_k) \cos\left(n \cdot \frac{k-0.5}{K} \cdot \pi\right), \tag{1}$$

where c_n is the nth MFCC coefficient, k is the number of Mel frequency bands, S_k is Mel-scaled power spectrogram of the signal, and n is the index of the MFCC coefficient (e.g., $n = 0, 1, \ldots, N-1$). We used $N = 40$, $K = 128$, and an FFT window length of 2048 samples, with 512 samples between successive frames. We tested different numbers of MFCCs, and $N = 40$ coefficients resulted in the best model accuracy results. This value was selected empirically.

Before building the model, we apply the T-distributed Stochastic Neighbor Embedding (t-SNE) technique to visualize our high-dimensional data in a 2D space. The input of the t-SNE model are the 40 MFCCs features calculated as explained above. Results are shown in Fig. 2. We can appreciate that the two classes (1-void and 0 no-void) are clearly distinct. It is then expected that we can find an ML model to perform automatic classification with high accuracy.

In order to select the best model that meets the requirements presented before, we have built four different ML models to compare the performance of the classification task.

- Logistic regression (LR): One main advantage of this model is that it is considerably lightweight.
- Support Vector Machine (SVM): this model uses a subset of training points in the decision function (called support vectors), which makes the model memory efficient. For the kernel functions, we selected a Radial Basis Function (RBF) kernel since it is the most widely used kernel due to its similarity to the Gaussian distribution.
- Random Forest (RF): this model fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. One main advantage of this model is that they are robust classifiers and they decrease the importance of features already duplicated by other features. We build a Random Forest classifier with 1000 estimators.
- Gradient Boost Classifier (GBC): It is an additive ensemble of a base model whose error is corrected in successive iterations (or stages) by the addition of Regression Trees which correct the error of the previous stage. One main advantage of this type of model is that it allows for the optimization of arbitrary differentiable loss functions.

4.3. DL model: transfer learning

Previous works have shown the use of Transfer learning (TL) to build robust, domain specific models. In this work we leverage the pre-trained YAMNet model [20] to classify audio clips as either voiding (1) or non-voiding (0) events. YAMNet is a convolutional neural network that utilizes the depthwise-separable convolution architecture of MobileNetV1 [21]. The model accepts data in the form of the log-mel-spectogram of 16 kHz single channel audio with a duration of 0.96s. The model converts these audio spectogram patches into a 1024-dimensional embedding. In the original YAMNet model, these embedding are passed to a single logistic layer to derive the 521-class output scores. In this work, the 521-class fully connected output layer is replaced by a fully connected 512 neuron layer with ReLu activation followed by another dense layer with 2 output neurons. The output of the model classifies audio events as either voiding (1) or non-voiding (0) events. We train on the loss using an Adam optimizer for 200 epochs with early stopping. The model is fed 0.96 clips of audio data, each of which corresponds to a voiding or non-voiding event.

5. Results and discussion

This section evaluates the performance of the models presented before, both the four ML models and the DL model.

5.1. Models performance evaluation

To evaluate the performance of the five classification models presented in the previous section, we use stratified k-fold validation to ensure that each fold of dataset has the same proportion of observations with a given label. The value $k = 5$ is used. For each model, we measure the following metrics: classification accuracy, False Positive Rate (FPR), and False Negative Rate (FNR).

The FPR provides information about what proportion of the class 1 got incorrectly classified, that is, no-void events classified as void events. The FNR provides information of what proportion of the class 0 got incorrectly classified, that is, void events classified as no-void events. It is important to note that regarding the end application of our model, it is desired to have a relatively low number of FNR, so that we decrease the probability of missing the recording of a voiding event. Results of the evaluation data are shown in Table 2. Results show that the GBC model provides the highest classification accuracy while providing a low FNR, with a value of 1.23%. The SVC model provides the lowest FNR, but the classification accuracy is significantly lower than the rest of the models.

The superior performance of the MFCC-GBC combination in our study compared to the DL model can be attributed to the feature engineering we used. MFCCs are well-established features in audio signal processing, particularly for tasks involving human or machine sound recognition. These features are designed to capture perceptually

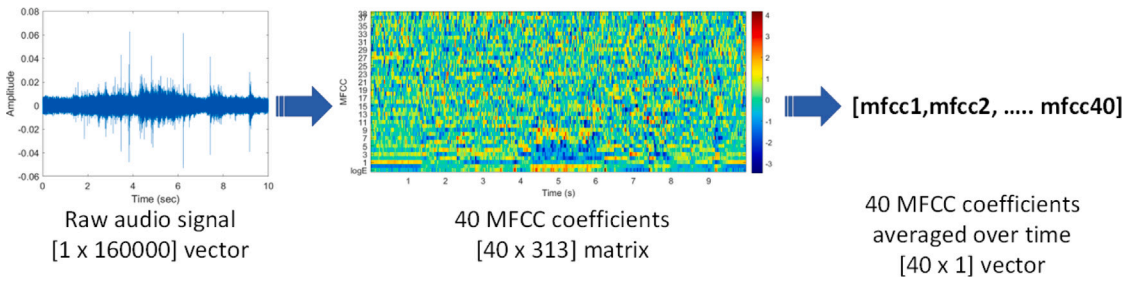

Fig. 3. Feature extraction pipeline for a 10-second-long audio clip for training the ML models.

Table 2
Models evaluation on the test data with stratified k-fold validation, $k = 5$.

	LR	SVC	RF	GBC	DL
Accuracy	90.49%	88.80%	92.39%	95.63%	92.04%
FPR	15.51%	22.61%	14.35%	7.68%	1.39%
FNR	3.84%	0.41%	1.23%	1.23%	16.88%

relevant information, effectively reducing the dimensionality of the data while preserving critical auditory characteristics. This inherent optimization aligns well with the task at hand. The CBG model performs well with high-quality input features and avoids some of the pitfalls of deep learning models, such as sensitivity to hyperparameter tuning and extensive computational requirements.

Next we present the receiver operating characteristic curve (ROC) along with the area under the curve (AUC) for the GBC and RF model, the two best performing models regarding the results of Table 2. The ROC curve evaluates their performance by plotting the TPR against the FPR at various threshold levels. The ROC metric was chosen as it is scale-invariant. Results are shown in Fig. 4. The two figures illustrate the ROC curve represented as a thick blue line for the Gradient Boosting Classifier (Fig. 4a) and the Random Forest classifier (Fig. 4b) across the 5-fold cross-validation setup. Regarding the GBC, the mean Area Under the Curve (AUC) of 0.98 ± 0.02, indicating excellent classification performance. The shaded region around the mean ROC curve represents the ±1 standard deviation, showing the robustness of the model across different folds. All individual folds achieved AUCs between 0.95 and 1.00, highlighting minimal performance variability. Similarly, the ROC curves for each of the five folds are displayed, with the model achieving high AUC scores. The mean ROC curve is also represented by a thick blue line, with a slightly higher mean AUC of 0.99 ± 0.01. The shaded area shows a tighter standard deviation compared to GBC, indicating even greater consistency in model performance across folds. Individual folds achieved AUCs close to 1.00, suggesting the Random Forest classifier slightly outperformed the Gradient Boosting Classifier in this classification task. Both models achieved near-perfect AUC scores, with a value close to 1, which means that they have a high measure of separability: prediction of void (or 1) audio clips as void, and no-void audio clips (or 0) as no-void.

5.2. Timing evaluation

This section compares the five different classification models in terms of the inference time, for three different embedded systems. The main hardware characteristics of the embedded systems evaluated are shown in Table 3. All devices are running RPi OS, but they all have different hardware characteristics. The RPi 4 B presents the highest computational power, but it also has the highest footprint and cost. The RPi Zero W is the cheapest one, but it does not support the 64-bits operating system. The RPi Zero 2 W presents the same size as the RPi Zero W, but with more computation power. Thus, the RPi Zero 2 W presents a good compromise between cost, footprint, and computational power.

Table 3
Comparison of three different RPi models.

Model	Zero W	Zero 2 W	4B
Rpi OS	32 bits	64 bits	64 bits
Architecture	ARM	ARM	ARM
	v6	Cortex-A53	Cortex-A72
Processor	1 GHz	1 GHz	1.5 GHz
RAM	512 MB	512 MB	4 GB
Footprint(mm)	65 × 30.5	65.6 × 30	85.6 × 56.5
Cost	10.44 €	18.90 €	74,10 €

Table 4
Timing evaluation of extracting the 40 MFCC features for a 10-seconds-long audio sample with the different embedded hardware platforms.

	RPi Zero W	RPi Zero 2 W	RPi 4
feature extraction (sec)	0.836	0.122	0.055

Table 5
Machine Learning models size.

Device	LR	SVM	RF	GBC	DL
Model size (KB)	2	92	319	73	2891

First we evaluate the time required for each model to extract the features that will be used to run inference. Results are shown in Table 4. These results show that the RPi Zero 2 W is more than 6 times faster than then RPi Zero, and only about 2 times slower than the RPi 4.

Next we evaluate and compare the inference time for the different models given an audio clip of 10 s long. The inference time is measured as the time needed by the program to extract the features and to return the classification result (0 or 1) for a 10-second-long audio clip collected with the microphone. Results are shown in Fig. 5. In all considered baselines, the model is uploaded just once and kept in memory. The DL model was only executed on the RPi 4 device since it needs a DRAM memory (main program memory) capacity higher than 512MB, which is not available on the other two RPi models.

Results show that for a particular embedded device, the ML model takes the longest time to run inference, while the LR model results in the shortest time. These results have a linear relationship with the model size shown in Table 5: lighter models have a lower inference time. Analyzing the traditional ML models, the RF is the slowest, followed by the GBC. Comparing the three different embedded devices, Fig. 5 shows that the RPi Zero W (gray color) is considerable slower than the other two devices for any given model.

5.3. Discussion

This work presented a new strategy that can facilitate the collection of sound flowmetries at home: provide the user with an embedded device that is capable of automatically detecting and recording their void events, with no user intervention. This approach aims to solve the problems related to current mobile app-based systems and those

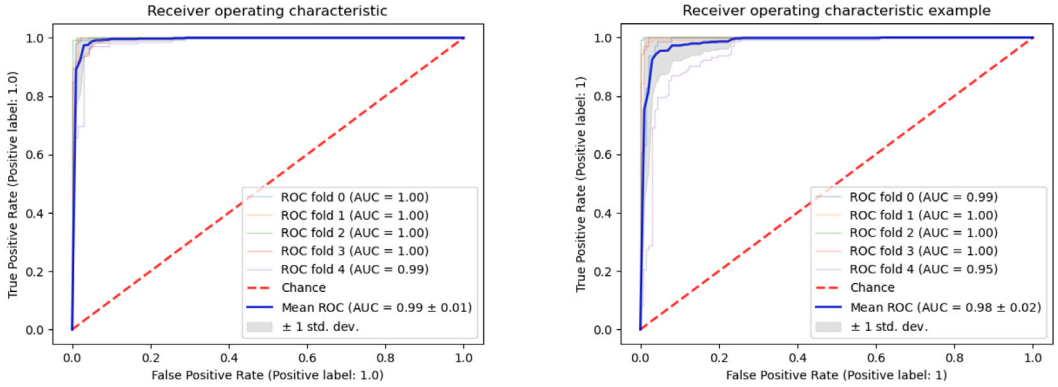

Fig. 4. ROC curve for the (a) GBC and (b) RF classifiers while performing the stratified-k-fold validation, with $k = 5$.

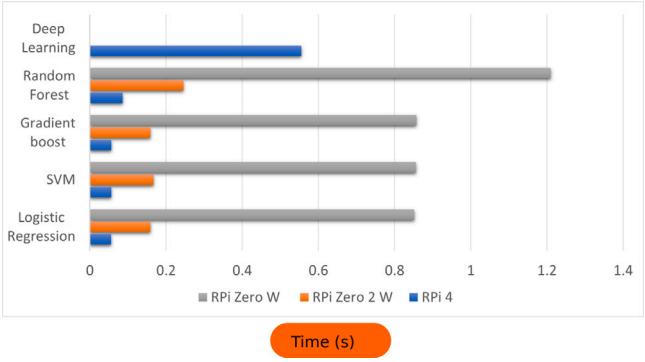

Fig. 5. Comparison of the inference time given a 10-seconds-audio clip, on three different embedded devices, for each classification model. x axis label shows time in seconds.

associated with standard flowmeters. Also, it can enable the collection of events at nighttime, since users typically forget to activate any device while voiding at nighttime. To the best of our knowledge, there is no previous work that solves this problem.

In the realm of classification algorithms, there is currently no existing model designed to predict void events based on audio signals. One widely utilized algorithm, Yamnet [20], adept at classifying audio events into 521 categories, lacks a specific class for void events among its 512 labels. When Yamnet is employed for inference using void events as input, the model frequently misclassified the sound as "Water", "Drip", "Liquid", or "Splatter". Notably, a comparable output is observed when Yamnet is applied to infer other toilet-related sounds such as hand-washing, flushing, and showering. Given these limitations, it becomes evident that there is a necessity to develop a new model tailored to void event classification and to curate an appropriate dataset for this purpose.

To automatize the detection and collection of voiding events, the use of traditional machine learning algorithms is justified since they achieve a high level of accuracy, as in such cases, the additional complexity of more advanced algorithms may not provide significant additional benefits. With the aim of facilitating replicability, we made public our GitHub repository where the collected data set is publicly available.

To extract the features from the audio data, we obtained the MFCCs. MFCCs were chosen as features due to their dimensionality reduction capabilities, noise robustness, statistical properties, and historical success in sound classification models. Results show that the GBC model provides the highest classification accuracy while providing a low FNR, with a value of 1.23%. The SVC model provides the lowest FNR, but the classification accuracy is significantly lower than the rest of the models. The GBC and RF model, the two best performing models, show an AUC value close to 1, meaning that they have a high measure of separability.

6. Conclusion

This work presented Auto-Flow, a proof-of-concept acoustic RPi-based platform that runs a novel ML classification model to automatically detect and record voiding events. This platform aims to address the problems associated with most current in-clinic uroflowmetry tests, where users experiment in an unnatural situation when they are required to void "on-demand". We have developed a low-cost platform that automatically detects and records voiding events at home that can be easily incorporated into a normal daily routine requiring minimal or even no human intervention. Results show that the GBC classifier, using 40 MFCC coefficients as input features, obtained from the audio clips, obtains a 95.63% classification accuracy to distinguish between void and no-void acoustic events. Future work will look at extracting the flow rate once the void event has been detected and recorded.

Overall, this work demonstrates the potential for the use of a low-cost embedded device in the assessment of voiding dysfunction, to deliver more personalized and effective health care at home with less waste of time and resources, in particular in rural or less developed areas where access to a urology specialist is more difficult.

Funding

Laura received funding as Juan de la Cierva Incorporation Fellow from the Spanish Ministry of Economy and Competitiveness, Spain (IJC2020-045901-I). This research has been supported by the Spanish Ministry of Science, Innovation and Universities under the AGINPLACE project, Spain (PID2023-146254OA-C44).

CRediT authorship contribution statement

Laura Arjona: Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **Sergio Hernández:** Writing – review & editing, Software, Methodology, Formal analysis, Conceptualization. **Girish Narayanswamy:** Writing – review & editing, Supervision, Methodology, Investigation. **Alfonso Bahillo:** Writing – review & editing, Writing – original draft, Project administration, Conceptualization. **Shwetak Patel:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] J. Cambronero Santos, C. Errando Smet, Prevalencia de síntomas del tracto urinario inferior de llenado en pacientes varones que acuden a consulta de urología en España. La urgencia urinaria como predictor de calidad de vida, Actas Urológicas EspaÑolas 40 (10) (2016) 621–627, http://dx.doi.org/10.1016/j.acuro.2016.04.012, Publisher: Elsevier.

[2] S. Sinha, The use of uroflowmetry as a diagnostic test, Curr. Urol. Rep. 25 (1) (2024) 99–107, http://dx.doi.org/10.1007/s11934-024-01200-0.

[3] S.A.Z. Zoe S. Gan, Current state and future considerations for home uroflowmetry, Nat. Rev. Urol. 20 (9) (2023) 515–516, http://dx.doi.org/10.1038/s41585-023-00785-4.

[4] C.V. Comiter, E. Belotserkovsky, A novel mobile uroflowmetry application for assessing low urinary tract symptoms, in: Neurourology and Urodynamics, 2018, Publisher: International Continence Society.

[5] L. Arjona, L.E. Diez, A. Bahillo, A. Arruza-Echevarria, UroSound: A smartwatch-based platform to perform non-intrusive sound-based uroflowmetry, IEEE J. Biomed. Heal. Inform. (2022) 1, http://dx.doi.org/10.1109/JBHI.2022.3140590.

[6] M. Lazaro, L. Arjona, M. Jojoa-Acosta, A. Bahillo, Flow prediction in sound-based uroflowmetry, Scientific reports 15 (2025) 643, http://dx.doi.org/10.1038/s41598-024-84978-w.

[7] L. Zhou, J. Bao, V. Watzlaf, B. Parmanto, Barriers to and facilitators of the use of mobile health apps from a security perspective: Mixed-methods study, JMIR MHealth UHealth 7 (4) (2019) http://dx.doi.org/10.2196/11223.

[8] M. Jung, S. Chi, Human activity classification based on sound recognition and residual convolutional neural network, Autom. Constr. 114 (2020) 103177, http://dx.doi.org/10.1016/j.autcon.2020.103177, URL: https://www.sciencedirect.com/science/article/pii/S0926580519307563.

[9] W. Wang, F. Seraj, P.J.M. Havinga, A sound-based crowd activity recognition with neural network based regression models, in: Proceedings of the 13th ACM International Conference on Pervasive Technologies Related to Assistive Environments, PETRA '20, Association for Computing Machinery, New York, NY, USA, 2020, http://dx.doi.org/10.1145/3389189.3389196.

[10] Z. Mnasri, S. Rovetta, F. Masulli, Anomalous sound event detection: A survey of machine learning based methods and applications, Multimedia Tools Appl. 81 (4) (2022) 5537–5586, http://dx.doi.org/10.1007/s11042-021-11817-9.

[11] S. Hernandez, L. Arjona, G. Narayanswamy, A. Bahillo, S. Patel, AutoFlow github repository, 2024, https://github.com/DeustoTech/AutoFlow.

[12] Seed Studio, ReSpeaker 2-Mics Pi HAT, 2024, https://wiki.seeedstudio.com/ReSpeaker_2_Mics_Pi_HAT/.

[13] K. Hosny, A. Magdi, A. Salah, O. Elkomy, N. Lashin, Internet of Things applications using Raspberry-Pi: a survey, Int. J. Electr. Comput. Eng. 13 (2023) 902–910, http://dx.doi.org/10.11591/ijece.v13i1.pp902-910.

[14] E. Microelectronic, EMBC22 beacon, 2024, https://www.emmicroelectronic.com/product/beacons/embc22.

[15] J.F. Gemmeke, D.P.W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R.C. Moore, M. Plakal, M. Ritter, Audio set: An ontology and human-labeled dataset for audio events, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2017, pp. 776–780, http://dx.doi.org/10.1109/ICASSP.2017.7952261.

[16] K.J. Piczak, ESC: Dataset for environmental sound classification, in: Proceedings of the 23rd Annual ACM Conference on Multimedia, ACM Press, 2015-10-13, pp. 1015–1018, http://dx.doi.org/10.1145/2733373.2806390, URL: http://dl.acm.org/citation.cfm?doid=2733373.2806390.

[17] M.K. Gourisaria, M.K. Gourisaria, R. Agrawal, M. Sahni, P.K. Singh, Comparative analysis of audio classification with MFCC and STFT features using machine learning techniques, Discov. Internet Things 4 (2024) http://dx.doi.org/10.1007/s43926-023-00049-y, URL: https://www.sciencedirect.com/science/article/pii/S0926580519307563.

[18] M. Veeramanickam, A. Ingavale, V. Khullar, S.S. Tirth, B. Pandey, H.P. Singh, Identification of sounds using deep learning with MFCC features extraction, in: 2024 IEEE AITU: Digital Generation, 2024, pp. 60–64, http://dx.doi.org/10.1109/IEEECONF61558.2024.10585335.

[19] Z.K. Abdul, A.K. Al-Talabani, Mel frequency cepstral coefficient and its applications: A review, IEEE Access 10 (2022) 122136–122158, http://dx.doi.org/10.1109/ACCESS.2022.3223444.

[20] S. Hershey, S. Chaudhuri, D.P.W. Ellis, J.F. Gemmeke, A. Jansen, R.C. Moore, M. Plakal, D. Platt, R.A. Saurous, B. Seybold, M. Slaney, R.J. Weiss, K. Wilson, CNN architectures for large-scale audio classification, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2017, pp. 131–135, http://dx.doi.org/10.1109/ICASSP.2017.7952132.

[21] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, 2017, CoRR abs/1704.04861, arXiv:1704.04861.