Supporting Information

A Piezoelectric Smart Patch Operated with Machine Learning Algorithms for Effective Detection and Elimination of Condensation

Qian Zhang[a,b], Yong Wang[a,b,c], Tao Wang[a], Dongsheng Li[a], Jin Xie[a,*], Hamdi Torun[b], Yongqing Fu[b,a,*]

[a] The State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

[b] Faculty of Engineering and Environment, University of Northumbria, Newcastle upon Tyne NE1 8ST, UK

[c] Key Laboratory of 3D Micro/Nano Fabrication and Characterization of Zhejiang Province, School of Engineering, Westlake University, Hangzhou 310024, China

* Corresponding authors: xiejin@zju.edu.cn; richard.fu@northumbria.ac.uk

Figures

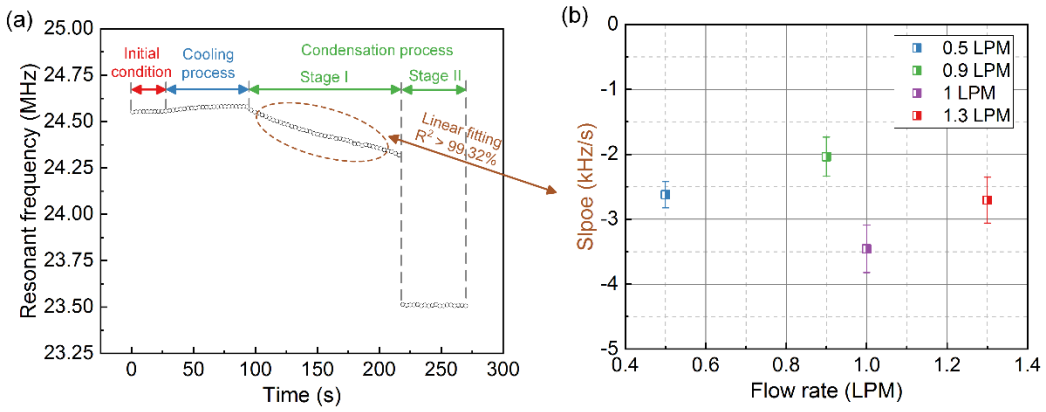

Figure S1. (a) The resonant frequency of the flexible device in the cooling and condensation processes. (b) The slope of the stage I of condensation process at four different flow rates. Error bars indicate the standard deviation (n = 3).

The effect of flow rate on the generation of condensation

As shown in the Fig. S1, experiments at different flow rates (0.5 LPM, 0.9 LPM, 1 LPM, 1.3 LPM) were conducted to investigate the influence of flow rate on the formation of condensation. The slope of stage I of condensation process is approximately proportional to the rate of condensation, which shows no significant decrease at a flow rate of 0.5 LPM compared to 1.3 LPM, as shown in Fig. S1(b). However, a flow rate of 0.9 LPM leads to a slower condensation rate, and a flow rate of 1.0 LPM results in a faster condensation rate. Therefore, it is summarized that the flow rate of gas pump affects the condensation rate, but the relationship between them is not a simple linear or polynomial function. The different condensation rates in the experiments of 0.9 LPM and 1 LPM are probably caused by the difference of flow fields in the chamber, which is determined by the structure of the chamber. On the other hand, because the gas flow rate is much larger than the rate of condensation, the humidity in the chamber keeps stable, which means the gas flow rate is "saturated". As a result, the condensation rates are comparable at gas flow rates of 0.5 LPM and 1.3 LPM.

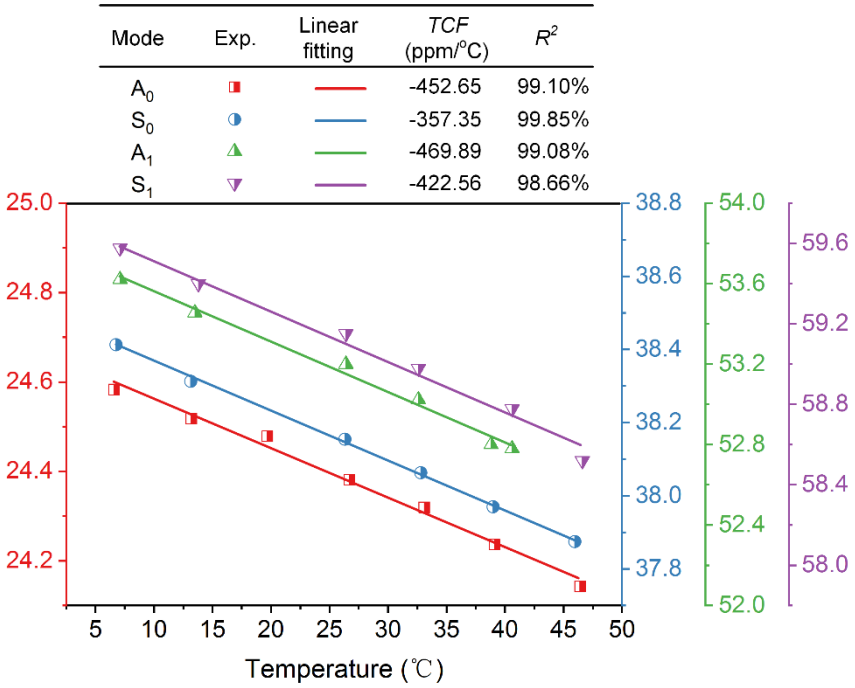

Figure S2. Temperature coefficients of frequency (TCF) of the A0, S0, A1, S1 modes of the flexible ZnO/Al device.

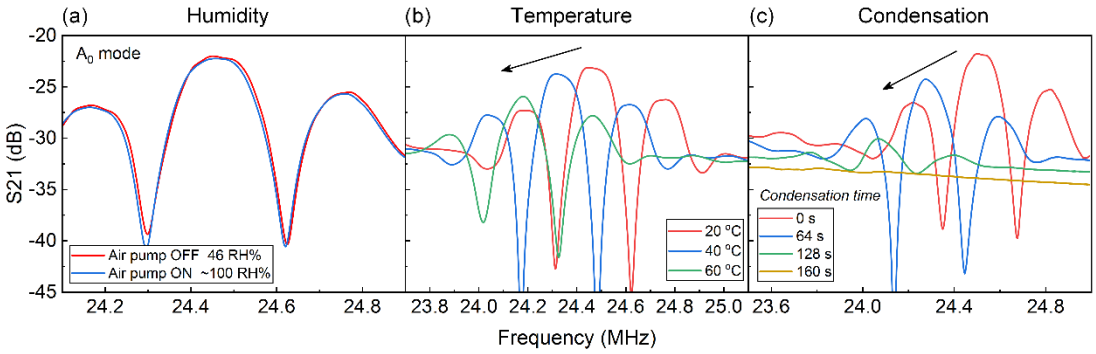

Figure S3. The changes of S21 spectra caused by variations of (a) humidity, (b) temperature and (c) condensation. Humidity has little effect on S21 spectrum, as shown in Fig. S3(a). The influence of heating on the S21 spectrum is similar to that of condensation, resulting in a decrease in resonant frequency and amplitude, as shown in Figs. S3(b) and S3(c). The similarity of the S21 spectra makes it suitable for effective classification using machine learning methods.

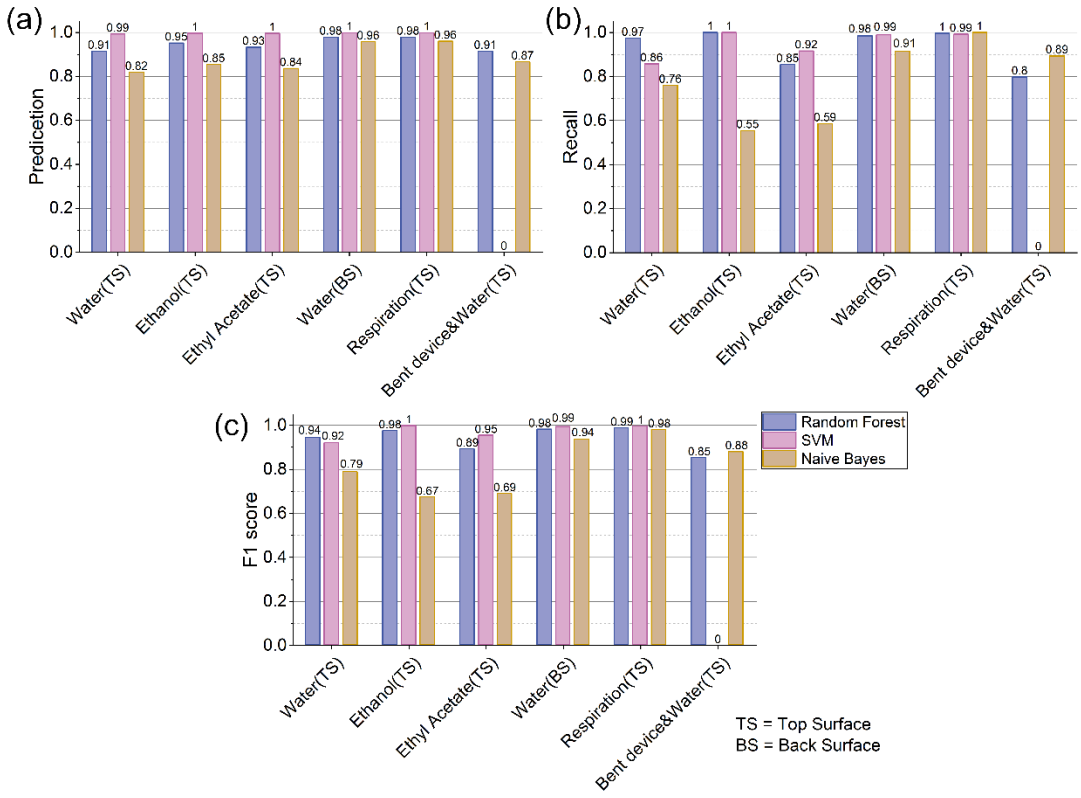

Figure S4. The classification performance, i.e., (a) *Prediction*, (b) *Recall* and (c) *F1 score* of six cases using random forest, support vector machine (SVM) and naive Bayesian model (NBM), respectively. The random forest algorithm shows comparable or outperforming *Precision*, *Recall* and *F1 score*. The SVM performs well in the first five cases, but does not work in the last one. The NBM performs worse than the random forest in nearly all the cases (I to V).

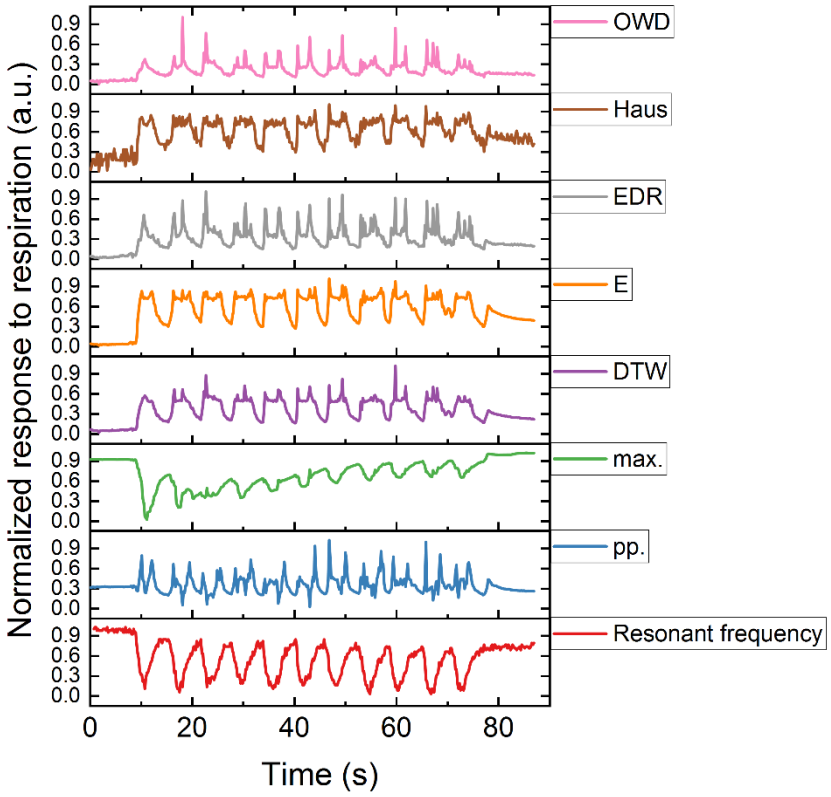

Figure S5. The normalized response to respiration in the form of the seven features and resonant frequency. The Hausdorff distance, Euclidean distance, DTW and resonant frequency are suitable as indicators used in respiration rate measurement.

Advantages compared with traditional method

Compared with the traditional measurement method on the basis of the shift of resonant frequency, the most significant advantage of the proposed detection method is that it can be used to distinguish the temperature change and generation of condensation, which can be applied for sensors with low power consumption and self-operation capability. For example, low sampling frequency (1 Hz) can be used for an inactive state, which is increased to 10 Hz for continuous measurement of respiration once detecting the generation of surface condensation. In addition, the selected features (Hausdorff distance, Euclidean distance, DTW) show relatively smaller static drift than that of the resonant frequency.

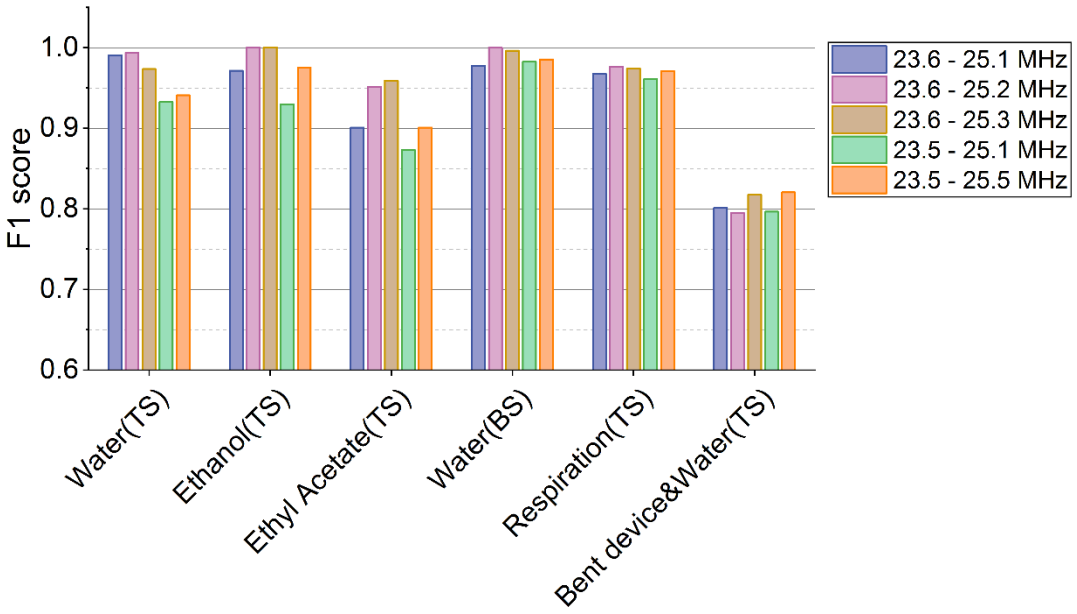

Figure S6. The classification results using five different frequency band selections, which show comparable *F1 scores*, indicating the robustness of the algorithm on the selection of frequency band. The sampling intervals are all set to 62.5 kHz in all the frequency bands.

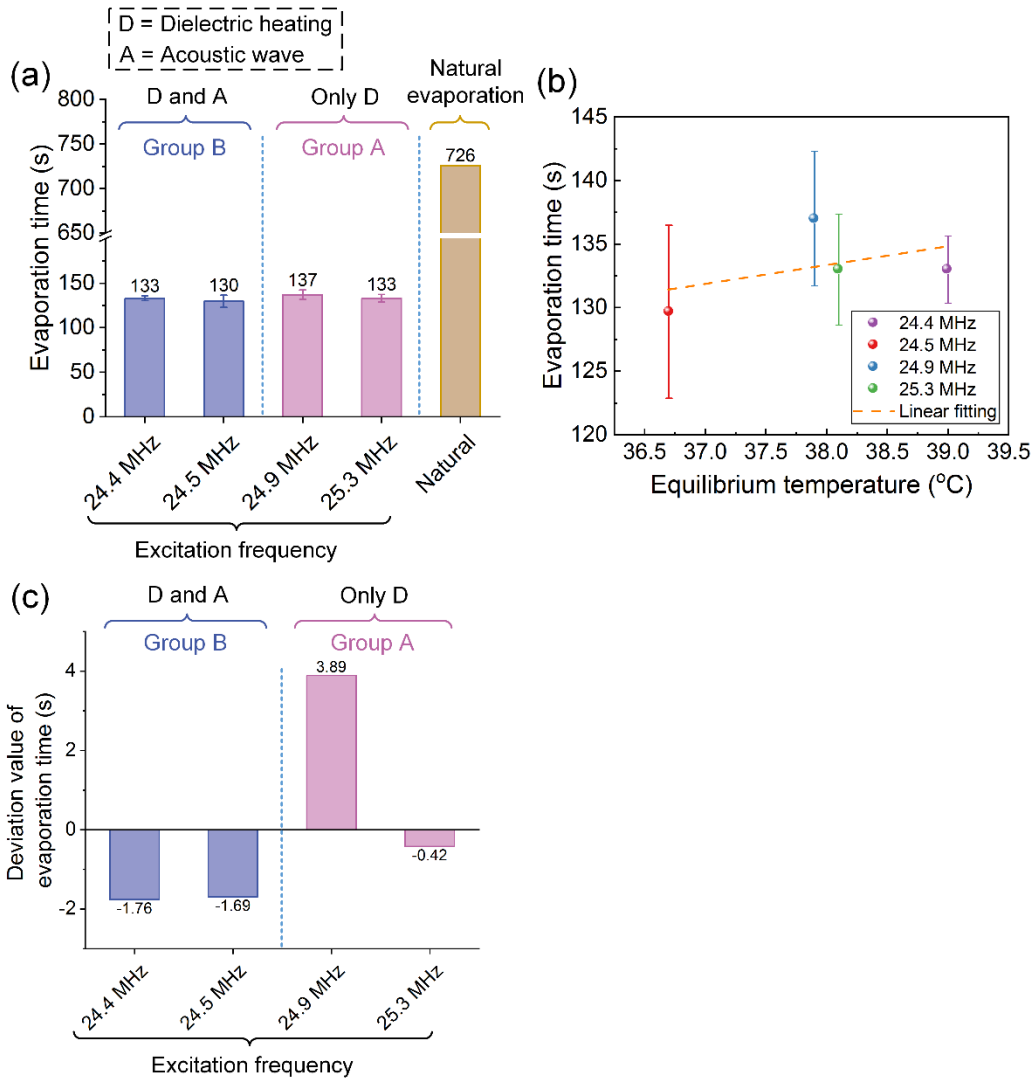

Figure S7. (a) Evaporation time of a deionized water droplet (~1 μL) at the center of the flexible device, which is excited by four sinusoidal signals with different frequencies (24.4 MHz, 24.5 MHz, 24.9 MHz and 25.3 MHz). Natural evaporation time is shown for comparison. (b) Evaporation time as a function of the equilibrium surface temperature. The linear fitting approximately indicates the contribution of temperature on accelerating evaporation in this temperature range (36.5°C to 39°C). (c) Relative evaporation time after removing the contribution of temperature, demonstrating that the acoustic waves do contribute to evaporation but less effective comparing to dielectric heating. Error bars in this figure indicate the standard deviation (n = 3).

Table

Table S1. The measured surface temperature readings of the flexible device after heating for 1 minute.

	Temperature (°C)			Average (°C)	Std. (°C)
Flat	34.3	35.2	35.2	34.9	0.5
Bent	32.7	33.7	34.9	33.8	1.1

To investigate the heating performance under bending condition, an acoustic wave device was excited by a sinusoidal power supply (25.3 MHz, ~ 31 Vpp and output impedance = 50 Ω) under flat and bent conditions separately at room temperature (25 °C), and the surface temperatures in both cases were measured after one minute. The radius of curvature after the bending was ~37.5 mm. The measurements were repeated for three times. As listed in Table S1, the results clearly show that bending does not significantly reduce the heating performance of the device.

Calculation of the seven features

The seven features used in this paper are peak-peak value, maximum value, dynamic time warping, Euclidean distance, edit distance on real signals, Hausdorff distance and one way distance. The calculation process of these scalar features can be given by

$$F^i = f^i\left(\boldsymbol{S}, \boldsymbol{S_0}\right), i = 1,2,...,7 \tag{S1}$$

where F^i is the value of i^{th} features, f^i is the corresponding function mapping the S21 parameters to the feature, $\boldsymbol{S}$ is the amplitudes of the S21 parameters in the selected frequency band (i.e., S21 vector) and $\boldsymbol{S_0}$ is the origin of the S21 vector which is defined as the S21 vector recorded at room temperature without condensation. Here is an introduction of the seven functions, i.e., f^i.

Peak-peak and Maximum value

The peak-peak value and maximum value are simple to calculate, which are exactly the peak-peak value and maximum value of the S21 vector.

Dynamic time warping (DTW)

The value of the third feature is defined as the distance between $\boldsymbol{S}$ and $\boldsymbol{S_0}$ using the dynamic time warping (DTW) technique. Assuming that $\boldsymbol{S}$ = [a_1, a_2, …, a_i, …, a_n], $\boldsymbol{S_0}$ = [b_1, b_2, …, b_j, …, b_n]. According to Ref. [1], the two sequences can be arranged to form a n-by-n grid, and a path can be generated starting from the point (a_1, b_1) and ending at the point (a_n, b_n). The warping path W = [w_1, w_2, …, w_k, …, w_m] and each w_k corresponds to a point $(i, j)_k$. To form the warping path, several restrictions are adopted, including

$$\left(i, j\right)_1 = \left(1,\ 1\right),\ \left(i, j\right)_k = \left(n,\ n\right) \tag{S2}$$

$$\left(i, j\right)_k = \left(i-1, j\right)_{k-1} or \left(i, j-1\right)_{k-1} or \left(i-1, j-1\right)_{k-1} \tag{S3}$$

With the distance defined as $\delta\left(i, j\right) = \left(a_i - b_j\right)^2$, the DTW distance can be obtained by

$$DTW\left(\boldsymbol{S}, \boldsymbol{S_0}\right) = \min\left[\sum_{k=1}^{m}\delta\left(w_k\right)\right] \tag{S4}$$

Euclidean distance

Before the calculation of Euclidean distance between $\boldsymbol{S}$ = [a_1, a_2, …, a_n] and $\boldsymbol{S_0}$ = [b_1, b_2, …, b_n], data preprocessing is conducted, including detrend and alignment. The detrend is accomplished by

$$\boldsymbol{S'} = \boldsymbol{S} - \mathrm{OLS}\left(\boldsymbol{S}\right), \mathrm{OLS} = \textit{ordinary least squares} \tag{S5}$$

where the OLS($\boldsymbol{S}$) indicates the linear fitting of $\boldsymbol{S}$. And $\boldsymbol{S_i''}$ can be obtained by shifting the $\boldsymbol{S'}$ by i elements

$$\boldsymbol{S_i''} = \begin{cases}\left[S'\left(i+1\right),...,S'\left(n\right),\boldsymbol{R}_i\right], \boldsymbol{R}_i = \left[S'\left(n\right),...,S'\left(n\right)\right]_{1\times i}, i \geq 0\\ \left[\boldsymbol{R}_i, S'\left(1\right),...,S'\left(n+i\right)\right], \boldsymbol{R}_i = \left[S'\left(1\right),...,S'\left(1\right)\right]_{1\times(-i)}, i < 0\end{cases} \tag{S6}$$

The Euclidean distance used in this paper can be expressed by

$$E\ \text{distance} = \min\left(\left\|\boldsymbol{S_i''} - \boldsymbol{S_0'}\right\|\right) = \min\left(\sqrt{\sum_{j=1}^{n}\left[\boldsymbol{S_i''}\left(j\right) - \boldsymbol{S'}\left(j\right)\right]^2}\right), i = -n+1, -n+2, ..., n-1 \tag{S7}$$

For the following 3 features, the data preprocessing including detrend and alignment of the peaks are also conducted before the calculation.

Edit distance on real sequence (EDR)

According to Ref. [2], the Edit distance on real sequence (EDR) is defined as the number of operations (insert, delete, or replace) that are needed to change $\boldsymbol{S}$ into $\boldsymbol{S_0}$, which takes the form

$$EDR(\boldsymbol{S},\boldsymbol{S}_0)=\min\left\{EDR\left(Rest(\boldsymbol{S}),Rest(\boldsymbol{S}_0)\right)+subcost, EDR\left(Rest(\boldsymbol{S}),\boldsymbol{S}_0\right)+1, EDR\left(\boldsymbol{S},Rest(\boldsymbol{S}_0)\right)+1\right\} \tag{S8}$$

$$subcost=\begin{cases}0, & if\ \left|S(1)-S_0(1)\right|\le tol\\ 1, & otherwise\end{cases} \tag{S9}$$

where the function *Rest*($\boldsymbol{S}$) is defined as the sub-matrix of $\boldsymbol{S}$ without the first element and the *tol* denotes the tolerance of matching (matching threshold). And *tol* = 0.1 is adopted in this paper.

Hausdorff distance

According to Ref. [3], the Hausdorff distance can be defined as

$$H(\boldsymbol{S},\boldsymbol{S}_0)=\max\left(h(\boldsymbol{S},\boldsymbol{S}_0),h(\boldsymbol{S}_0,\boldsymbol{S})\right) \tag{S10}$$

where

$$h(\boldsymbol{S},\boldsymbol{S}_0)=\max_{a\in\boldsymbol{S}}\min_{b\in\boldsymbol{S}_0}|a-b| \tag{S11}$$

One way distance (OWD)

In order to calculate the one way distance (OWD), 2-D curves $\boldsymbol{C}$ and $\boldsymbol{C}_0$ are generated from the vectors $\boldsymbol{S}$ and $\boldsymbol{S_0}$ by adding a dimension. For example, $\boldsymbol{S}$ = [a_1, a_2, …, a_i, …, a_n] and the corresponding $\boldsymbol{C}$ = {(k, a_1), (2k, a_2), …, (ik, a_i), …, (nk, a_n)}, where the ik is the manually defined dimension and can be regarded as the punishment on frequency differences. In order to set similar weights for the two dimensions, the value of k is determined by the following normalization

$$(n-1)k=\max_{a_i\in\boldsymbol{S}_0}(a_i)-\min_{a_i\in\boldsymbol{S}_0}(a_i) \tag{S12}$$

According to Ref. [4], the distance from a point p = (ik, a_i) to $\boldsymbol{C_0}$ is defined as

$$D_{point}(p,\boldsymbol{C}_0)=\min_{q\in\boldsymbol{C}_0}\left(D_{Euclid}(p,q)\right) \tag{S13}$$

where the D_{Euclid}() denotes the Euclidean distance. Then the OWD from the $\boldsymbol{C}$ to the $\boldsymbol{C}_0$ is defined as

$$D_{OWD}(\boldsymbol{C},\boldsymbol{C}_0)=\frac{1}{|\boldsymbol{C}|}\left(\sum_{p\in\boldsymbol{C}}D_{point}(p,\boldsymbol{C}_0)\right) \tag{S14}$$

The distance between the $\boldsymbol{C}$ and $\boldsymbol{C}_0$ is the average of the two OWDs

$$D(\boldsymbol{C},\boldsymbol{C}_0)=\frac{1}{2}\left(D_{OWD}(\boldsymbol{C},\boldsymbol{C}_0)+D_{OWD}(\boldsymbol{C}_0,\boldsymbol{C})\right) \tag{S15}$$

Random forest classifier

Random forest algorithm can be used effectively to rank different features or factors based on their importance. Therefore, it is adopted in order to find out which features are dominant in classification of various parameters whilst paying importance to the physical meaning of the behavior of the S21 spectra instead of trying to build only a statistical model for the device. For example, the maximum value, corresponding to the amplitude of the acoustic wave, is the most important one for classification, which indicates that the energy dissipation caused by condensation is an obvious feature. However, the peak-to-peak value is not important for classification, indicating that the minimum value is less significant. Therefore, the minimum energy transmission might not need to be paid much attention in this work.

The basic principle of the random forest algorithm lies in the training of an ensemble of decision trees (CART trees used here) with each tree growing individually [5-7]. The predicted classification result is obtained by a majority vote of all the decision trees, which takes the form

$$y = \begin{cases} 1 & if \ \frac{1}{N}\sum_{i=1}^{N} y_i > 0.5 \\ 0 & otherwise \end{cases} \tag{S16}$$

where y_i is the predicted result of the i^{th} decision tree and y denotes the estimation of the forest. The detailed random forest algorithm used in this paper is shown in the following **Algorithm 1**, which was finally realized by a MATLAB program. According to Ref. [5-6], the random forests do not overfit as more trees are added, but produce a limiting value of the generalization error. As a result, the number of trees N is only limited by the computing resources and can be chosen arbitrarily large, e.g., infinity. For simplicity of calculation, the number of trees N is 200 in calculation, the prediction of which has been proved to converge to an acceptable level. For each tree, a special training set is obtained by randomly selecting N points in the training set with replacement. Recommended by Ref. [6-7], the minimum leaf size is set as 1 and the number of randomly selected features, without replacement, used in each split is set to $p^{1/2}$, where p is the total feature numbers.

Algotirhm 1: random forest for classification

Input: Training set $\mathcal{T}_n = [\boldsymbol{X}_{n\times p}, \boldsymbol{Y}_{n\times 1}]$, Min leaf size $MLS = 1$, Number of trees $N = 200$, Test point $\boldsymbol{x}_{1\times p}$.

Output: Predicted classification of the random forest at $\boldsymbol{x}_{1\times p}$, denoted y.

for $i = 1, \dots, N$ **do**

  Select n points (rows) without replacement in the training set $\mathcal{T}_n$, generating a temporary set $\mathcal{S}_n = [\boldsymbol{X'}_{n\times p}, \boldsymbol{Y'}_{n\times 1}]$.

  Set a list $\mathcal{L}$ containing its only element $\mathcal{S}_n$, which is the root cell of the tree.

  Set $\mathcal{L}_f = \emptyset$ an empty list.

  while $\mathcal{L} \neq \emptyset$ **do**

    Let E be the first element of $\mathcal{L}$.

    if *E contain less than or equal to MLS, or if all* $X_i \in A$ *are equal* **then**

      Remove the cell E from the list $\mathcal{L}$.

      Add the cell E in the list $\mathcal{L}_f$.

    else

      Randomly select $q=floor(p^{1/2})+1$ features (columns), without replacement, from $\boldsymbol{X'}_{n\times p}$ to form a subset $\boldsymbol{X''}_{n\times q}$.

      Find the best split for $[\boldsymbol{X''}_{n\times q}, \boldsymbol{Y'}_{n\times 1}]$ by optimizing the CART-split criterion.

      Cut the cell E according to the best split, generating two resulting cells, E_1 and E_2.

      Remove the cell E from the list $\mathcal{L}$.

      Add the cells E_1 and E_2 in the list $\mathcal{L}$.

    end

  end

  Compute the estimate y_i at the point $\boldsymbol{x}_{1\times p}$ of the i^{th} decision tress classifier indicated by $\mathcal{L}_f$.

end

Compute the estimate y at the point $\boldsymbol{x}_{1\times p}$ of the random forest classifier via a majority vote according to Eq. S1.

The function *floor*() is defined as rounding toward negative infinity.

References

[1] Berndt, D. & Clifford, J. Using dynamic time warping to find patterns in time series. *Work. Knowl. Knowl. Discov. Databases* **398**, 359–370 (1994).

[2] Chen, L., Özsu, M. T. & Oria, V. Robust and fast similarity search for moving object trajectories. *roc. ACM SIGMOD Int. Conf. Manag. Data*, 491–502 (2005).

[3] Huttenlocher, D. P., Rucklidge, W. J. & Klanderman, G. A. Comparing images using the Hausdorff distance under translation. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* **1992-June**, 654–656 (1992).

[4] Lin, B. & Su, J. One way distance: For shape based similarity search of moving object trajectories. *Geoinformatica* **12**, 117–142 (2008).

[5] Breiman, L. Random Forests. *Mach. Learn.* **45**, 5–32 (2001).

[6] Biau, G. & Scornet, E. A random forest guided tour. *Test* **25**, 197–227 (2016).

[7] Liaw, A. & Wiener, M. Classification and Regression by randomForest. *R news* **2**, 18–22 (2002).