High-performance Edge-side Classification of Multi-modal Spectroscopic Data by Neural Network with Random Weights

Yinsheng Zhang, Zhengyong Zhang, Dian Rong, Yongbo Cheng, Xiaolin Qin, and Haiyan Wang

I. Introduction

A. Spectroscopic Profiling Technology

Spectroscopic profiling technology has been widely used by both manufacturers and governments to analyze and monitor the physio-chemical property of materials and products. Among these spectroscopic instruments, Raman spectrometry, MALDI-TOF-MS (matrix-assisted laser desorption/ionization time-of-flight mass-spectrometry), UV (ultraviolet) spectrometry, IMS (ion mobility spectrometry) and other rapid analytical devices, have become a promising alternative to traditional chromatography-based methods (e.g., GC-MS, LC-MS). Some advantages of these rapid analytical spectroscopic methods include, 1) less or no need for complex sample preprocessing; 2) require less sample amount; and 3) faster scan speed. More importantly, when combined with statistical methods, these instruments can improve their analytical accuracy with more data.

B. Remote Sensing and Edge Computing

In this era of IoT (internet of things), edge computing has been proposed to improve various remote sensing applications. Edge computing is an extension to traditional cloud computing, where end-point sensors (e.g., spectroscopic profiling instruments) are endowed with some computational power and artificial intelligence. In this way, the edge-side devices can act autonomously and independently, reducing the computation and communication burden of the central server.

In food/pharmacy companies, remote sensing by spectroscopic profiling instruments is a cost-effective way for both production-line monitor and government surveillance. However, one unique challenge is high-performance algorithms, which can suit the limited computational power and cost-effective edge-side hardware.

C. Fusion of Multi-modal Data

Another challenge in the spectroscopic profiling-based remote sensing is how to handle multi-modal data, which come from different spectroscopic instruments (e.g., RAMAN + UV) based on different physio-chemical principles. This phenomenon of multi-modality also exists in other domains. For example, in healthcare, physicians use multiple modalities, such as MRI (Magnetic Resonance Imaging), CT (Computed Tomography), microscope, DNA sequencing, etc., to give a comprehensive clinical judgment. Another example is the self-driving car, where different types of sensors are used together, e.g., video cameras, radar, ultrasonic sensors and LIDAR (Laser Imaging Detection and Ranging). Even in our daily life, we inference on the physical world with our multiple biological senses, i.e., vision, hearing, touching, etc.

In this paper, the fusion of remote sensing data from Raman and UV modalities is addressed.

D. Manuscript Structure

In the following manuscript, we will first introduce a specific dataset from spectroscopic profiling-based remote sensing. Then, we will propose a multi-modal spectroscopic data fusion and classification algorithm. Finally, we will conduct a case study on the dataset and compare our algorithm with peers.

II. Dataset and Problem Statement

A. Test Subjects

The test subject is radix astragali (astragalus root). Chinese name: 黄芪 (huang-qi). Radix astragali is a widely used medicinal herb in traditional Chinese medicine (TCM). It is often used to treat diabetes and cardiovascular diseases.

The radix astragali samples are provided by a TCM pharmaceutical manufacturer. The radix astragali is a main ingredient in one of their patent drugs.

The radix astragali samples come from four different provinces, i.e., Neimeng (north China), Sichuan (southwest China), Shanxi (northwest China) and Gansu (northwest China). Herbs from different regions could have slightly different medicinal effectiveness, due to different cultivation

Correspondence author: Haiyan Wang. This work was supported in part by the National Natural Science Foundation of China under Grant 91746202, 61806177, 71433006 and 61602217, and in part by the China Scholarship Council under Grant 201808330609.

Y.S. Zhang, D. Rong, and H.Y. Wang are with the School of Management and E-Business, Zhejiang Gongshang University, Hangzhou 310018, China. (e-mail: zhangys@illinois.edu; rongdian@zjgsu.edu.cn; njue2010@163.com)

Z.Y. Zhang and Y.B. Cheng are with the School of Management Science and Engineering, Nanjing University of Finance and Economics, Nanjing, 210023, China. (email: zyzhangnjue@126.com; ybc@nuaa.edu.cn)

X.L. Qin is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China. (email: qinxcs@nuaa.edu.cn)

conditions, climate, and environment.

The purpose of the case study is to discriminate the geographic source of radix astragali raw materials.

B. Instrument

Instruments: Both Raman and UV are used to profile the radix astragali samples. The Raman spectrometer used in this study is ProttezRaman-D3, manufactured by Enwave Optronics, US. The ultraviolet-visible absorption spectrum is generated by a T6 ultraviolet-visible spectrometer, manufactured by Purkinje General Instrument, China.

Procedures: Each radix astragali sample is first processed in a high-speed grinder for 10 min at 25,000 rpm. Every 3 g of the grinded powder is solved in 30 mL ethanol solution. Then, the mixture is stirred for 1 h at 100℃ with reflux. Let the mixture cool down and then filter it. Finally, test with Raman and UV spectrometers. The parameters for Raman are as follows. Laser wavelength: 785 nm. Laser power: 450mW. Exposure time: 5 s.

C. Dataset

Totally 160 radix astragali samples are tested. Each category (province) has 40 samples.

The Raman and UV spectra are organized into two CSV (comma-separated value) files. Each row is one sample. The first column is the Y label (0-Neimeng, 1-Sichuan, 2-Shanxi, 3-Gansu). The rest of the columns are Raman wavenumbers or UV wavelengths. Data file summary:

7044.txt - Raman
X meaning: Raman shift / wave number
X range: 100 ~ 4278 cm-1
Resolution: 2cm-1

7143.txt - UV
X meaning: wave length
X range: 200 ~ 800 nm
Resolution: 1 nm

Each Raman spectrum has 2091 dimensions, and each UV spectrum has 602 dimensions. Each dimension is also called an attribute (computer science term), a feature (a machine learning term), or a variable (a statistical term). In this paper, we use these terms interchangeably.

D. Problem Statement

The unique challenges in this research come from two aspects. 1) One challenge is related to the domain data. The spectroscopic data in this study is both multi-modal (Raman and UV) and highly dimensional (2091 and 602 dimensions), which require effective data fusion and dimensionality reduction techniques. 2) The other challenge comes from the endpoint hardware limitation. In remote sensing, the edge-side sensor uses cost-effective hardware (unpowerful processor, small memory, less cache, etc.), which has limited computation power. This requires specially tailed and high-performance algorithms.

Therefore, this study will propose a high-performance edge-side classification algorithm on the multi-modal spectroscopic dataset. The algorithm is based on NNRW (Neural Network with Random Weights). NNRW provides a faster training mechanism than GD (gradient descent)-based methods.

III. Theory and Method

A. Neural Network with Random Weights

The last two decades sees the renaissance of artificial neural network (ANN) models. Thanks to deep learning techniques, ANN can go deeper and solve more complex problems. The effectiveness of ANN is guaranteed by the universal approximation theorem (UAT). UAT states that a neural network with one hidden layer (e.g., a perceptron model), when properly parameterized, can approximate any continuous function on the compact subsets of R^n.

Traditionally, ANN is mostly trained by the back propagation (BP) method, which is based on GD (gradient descent). BP and GD are very successful in the machine learning field. They have been extensively used in training very deep ANN models. However, they also have several shortcomings, including: 1) may stuck in local minima, 2) long convergence time due to gradient-descent iterations, and 3) sensitive to hyper-parameters (e.g., learning rate, batch size, and initialization strategy).

As a complement to BP-based ANN, NNRW has been proposed. In NNRW, all (or part of) the weights are randomly assigned and don't need to be finetuned or trained by gradient descent-based iterations. In the next part, we will introduce the popular RVFL (Random Vector Functional-Link) family. RVFL used a mixed feed-forward ANN architecture (Fig. 1). It has an input layer, an output layer, and a hidden layer (a.k.a., enhancement layer).

Let's denote $\mathbf{\Theta^{(1)}}$ as the weight matrix between the input and hidden layers. $x = [x_0\ x_1\ x_2\ \ldots]^T$ is the input vector. x_0 (always equals to 1) is a constant node added for the bias parameter. $a^{(2)} = [a_1^{(2)}\ \ a_2^{(2)}\ \ldots]^T$ is the activation vector of the hidden layer nodes. $z^{(3)} = [z_1^{(3)}\ \ z_2^{(3)}\ \ldots]^T$ is the output vector.

$\boxed{\neq}$ in the hidden layer means it uses a non-linear activation function (e.g., sigmoid). The basic assumption of RVFL is that $\mathbf{\Theta^{(1)}}$ is independent of the dataset, and be set randomly, usually using a uniform random distribution on (-1, 1). Following the forward propagation rule, we have:

$$a^{(2)} = g(\Theta^{(1,2)}x) \quad (1)$$

Then, we concatenate x and $a^{(2)}$ as one vector v.

$$v = \left[x_{0?} x_1\ x_2 \ldots a_1^{(2)}\ a_2^{(2)} \ldots \right]^T = \begin{bmatrix} x \\ g(\Theta^{(1)}x) \end{bmatrix} \quad (2)$$

Define $\mathbf{\Theta^{(2)}}$ as the weight matrix between v (input layer and hidden layer combined) and z (the output layer). It should be noted that, in RVFL the output layer doesn't use any activation function. v is directly connected to the output nodes. Therefore,

$$z^{(3)} = \Theta^{(2)}v = \Theta^{(2)} \begin{bmatrix} x \\ g(\Theta^{(1)}x) \end{bmatrix} \quad (3)$$

Because the random weight matrix $\boldsymbol{\Theta^{(1)}}$ is already known, we only need to solve $\boldsymbol{\Theta^{(2)}}$.

The above equations (1)(2)(3) are related to one sample *x* (a *n*-dimensional column vector). We can further extend the equations to the entire dataset, e.g., we use capitalized *X* (an $m \times n$ matrix, *m* is the sample size, *n* is the feature number) to denote the dataset. Then, we have:

$$V = [X \quad g(X\Theta^{(1)})] \quad (4)$$

$$Z = V\Theta^{(2)} = [X \quad g(X\Theta^{(1)})]\Theta^{(2)} \quad (5)$$

We can make a transformation to equation (5):

$$V^T Z = V^T V\Theta^{(2)} = (V^T V)\Theta^{(2)} \quad (6)$$

$$(V^T V)^{-1} V^T Z = (V^T V)^{-1}(V^T V)\Theta^{(2)} = \Theta^{(2)} \quad (7)$$

$(V^T V)$ in equation (6) is a square matrix, so we can calculate its inverse $(V^T V)^{-1}$ in equation (7).

Finally, we have:

$$\Theta^{(2)} = (V^T V)^{-1} V^T Z = V^+ Z \quad (8)$$

Denote $V^+ = (V^T V)^{-1} V^T$. V^+ is the pseudo-inverse matrix, a.k.a., the Moore–Penrose inverse. It is a generalization of the ordinary inverse matrix. V^+ computes the L2-norm (least squares) best fit solution to the linear equation system $V\Theta^{(2)} = Z$.

One advantage of RVFL is that we can directly compute the final solution, without any gradient-based iterations. Another advantage is that RVFL combines both linear (provided by the direct link between *v* and *z*) and non-linear (provided by the sigmoid activation in the hidden layer) capabilities. This improves the model's fit power and generalization capability.

One major disadvantage of RVFL is that, it cannot handle high-dimensional dataset efficiently. Let *n* be the feature number of *x*, *l* be the hidden layer node number. Then $V^T V$ will be a (*n*+*l*) × (*n*+*l*) matrix. Calculating its inverse can have a huge computational cost.

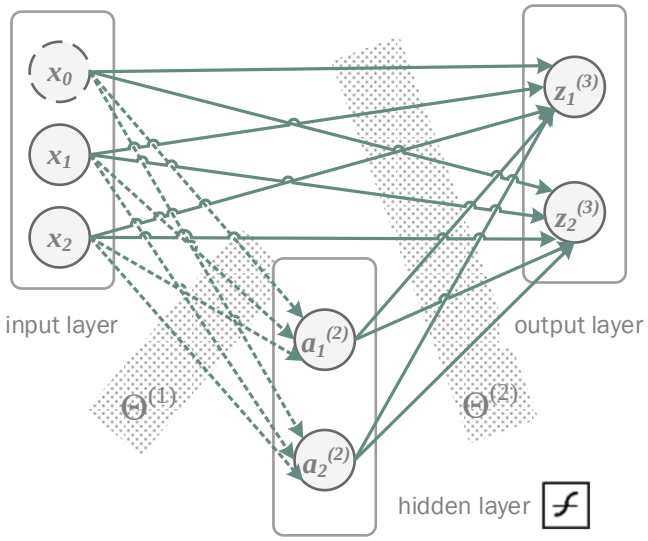


Fig. 1. Architecture of Random Vector Functional-Link (RVFL) Network. The node connections covered by $\boldsymbol{\Theta^{(1)}}$ and $\boldsymbol{\Theta^{(2)}}$ are indicated by the dotted-shadow bands.

Besides RVFL, recent studies on WANN (Weight Agnostic Neural Network) and neuro-evolution are also closely related to NNRW. Based on UAT, WANN tries to find an ANN architecture, which can greatly outperform pure chance even with random weights. Neuro-evolution tries to find the best architecture candidate by searching the network topology space by algorithms such as GA (genetic algorithm).

B. Multi-modal Spectroscopic Data Classification using NNRW

In addressed to the two aforementioned challenges (one is multi-modal data fusion and dimensionality reduction; the other is high-performance edge-side algorithm), a multi-modal spectroscopic data classification pipeline is proposed (Fig. 2). The pipeline contains the following procedures.

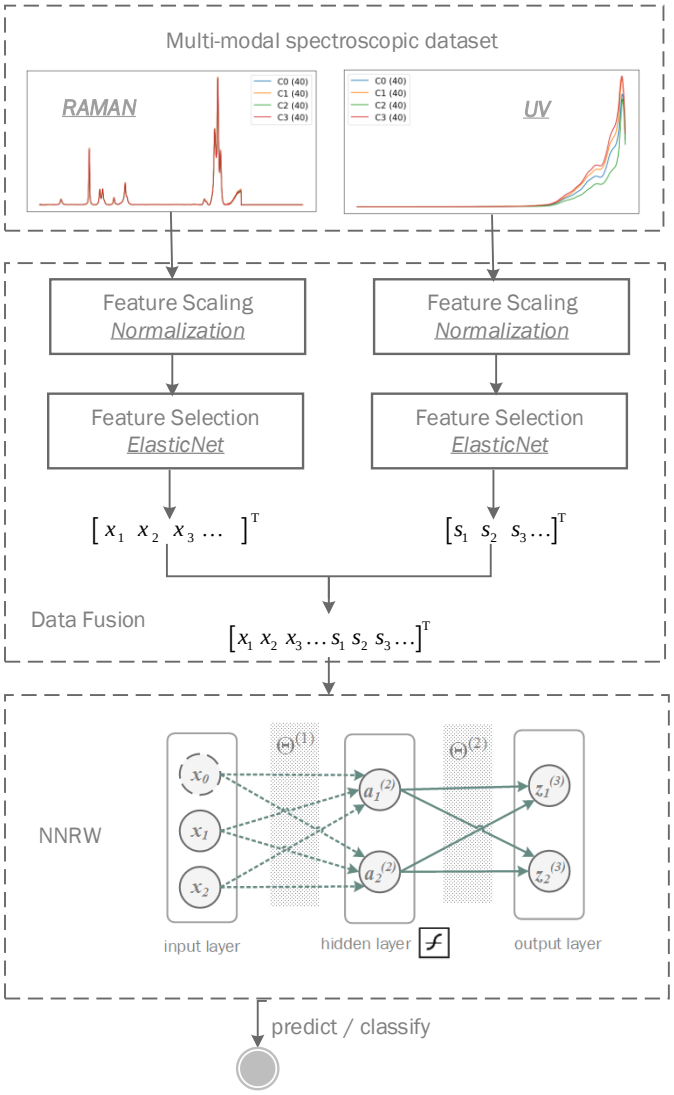


Fig. 2. Multi-modal Spectroscopic Data Classification Workflow using NNRW

1) *Feature Scaling*

In this pipeline, feature scaling is required by the following feature selection procedure, based on the Elastic Net [1]. Feature selection methods, such as LASSO (least absolute shrinkage and selection operator) [2] and Elastic Net, are essentially based on linear regression. The regression coefficient (absolute value) of each feature represents its importance. For spectroscopic profiling data, the magnitudes of the features vary significantly, and determine the regression coefficients. Therefore, it is necessary to rescale the features to a similar range. In this study, variable normalization is used for feature scaling: $x' = \frac{x - \mu}{\sigma}$.

2) *Feature Selection*

In this pipeline, feature selection serves several purposes:

- Reduce data dimensionality and remove irrelevant features. This helps reduce the overfitting risk of the dataset and improve model generalization ability.
- Improve the explainability of the algorithm. By selecting only a few important features, the domain expert can better interpret the chemical meanings of the spectroscopic data.
- Improve the performance of the NNRW model. In NNRW, the calculation of the inverse matrix $(V^T V)^{-1}$ is quite expensive. The matrix size is decided by the feature numbers.

For the above three reasons, feature selection is highly recommended for spectroscopic data, such as Raman (2091 features) and UV (602 features). Other high-dimensional spectroscopic data, such as MALDI-TOF (20k ~ 40k features), can also benefit from feature selection.

The feature selection algorithm in this study is Elastic Net. Elastic Net is a linear regression model, which tries to minimize the following cost function:

$$J(\Theta) = \underbrace{\frac{1}{m}\sum_{i=1}^{m}[\sum_{j=0}^{n}(\theta_j x_j^{(i)}) - y^{(i)}]^2}_{MSE} + \underbrace{\lambda_1 \sum_{j=1}^{n}|\theta_j|}_{L1\ penalty} + \underbrace{\lambda_2 \sum_{j=1}^{n}\theta_j^2}_{L2\ penalty}$$

The cost function uses both L1 and L2 penalties for regularization, which equals to a combination of LASSO and ridge regression. The L1-norm penalty has a sparse effect, which tends to generate more zero-valued coefficients.

3) *Neural Network with Random Weights*

After feature selection, only a small subset of features is kept for each modality, e.g., Raman or UV. Features in these subsets have the most discriminative power among the different categories (Y labels). These selected features are then concatenated into one long feature vector.

The final step of the pipeline is to train a classifier based on the feature vectors after data fusion. As discussed before, the NNRW is a promising model to implement high-performance edge-side classification algorithm. Rather than the full-fledged RVFL model, we choose the more efficient variant - ELM (Extreme Learning Machine). As seen in Fig. 3, ELM is a simplified version of RVFL, where the direct link between *x* and *z* is removed. According to the theory of ELM, the weight matrix $\mathbf{\Theta^{(1)}}$ between the input and hidden layers need not to be learned, and is independent of specific datasets. By UAT, with enough hidden nodes, ELM can fit any continuous function in the specific domain.

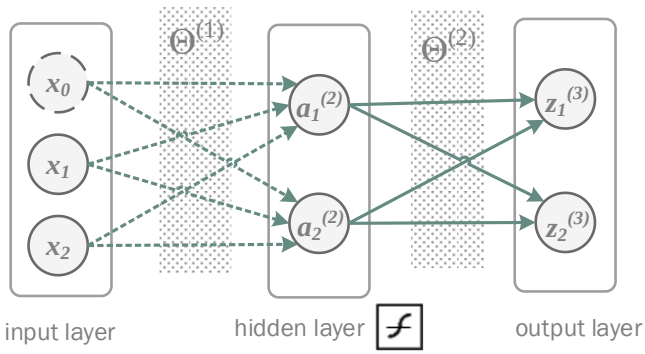


Fig. 3. Architecture of Extreme Learning Machine (ELM).

To train/fit ELM, the weight matrix $\mathbf{\Theta^{(1)}}$ is first randomized, e.g., from a uniform random distribution. Then, we solve the weight matrix $\mathbf{\Theta^{(1)}}$ by equation (8): $\Theta^{(2)} = (V^T V)^{-1} V^T Z = V^+ Z$.

Concerning this equation, $V = [X \;\; g(X\Theta^{(1)})]$ for a typical RVFL, whereas $V = g(X\Theta^{(1)})$ for ELM.

IV. Case Study and Experiment

A. Result of Our Method

Based on the proposed algorithm pipeline (Fig. 2), the multi-modal spectroscopic data of Raman and UV are analyzed.

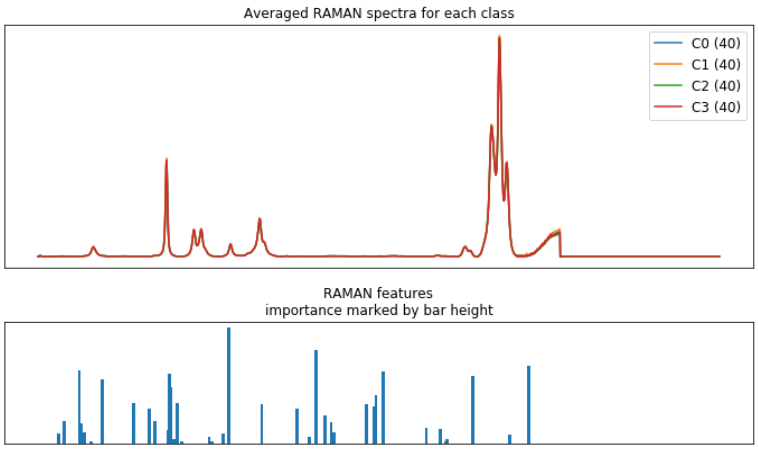
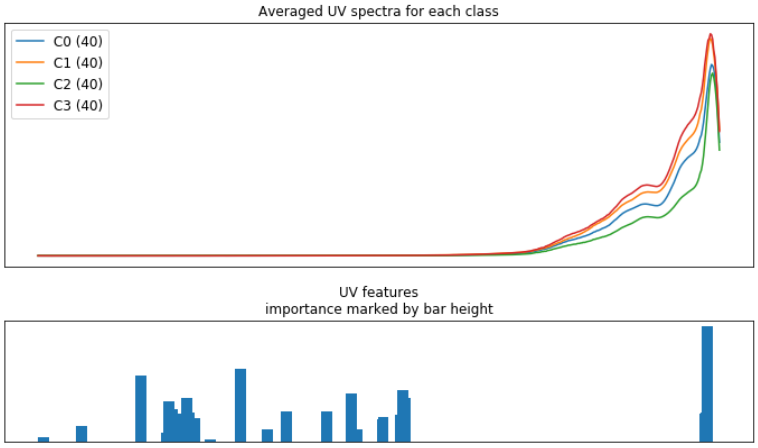
1) Feature Selection and Data Fusion

Table 1 shows the feature selection result from Elastic Net. The averaged spectrum and feature importance chart are provided. After the Elastic Net regression, 48 and 32 features are selected (with non-zero regression coefficients) from the Raman and UV dataset respectively. Chemical interpretations of the top-n selected features are also provided. The final concatenated vector has 80 (48+32) features.

2) Classification

This case study uses ELM for classification. ELM has one hyper-parameter: the hidden node count (denote as *L*). According to UAT, the fitting power and complexity of the

Table 1. Feature selection result and chemical explanation for the top-5 selected features (Raman shifts / UV wavelengths)

Modality	Feature importance chart (bar height = coefficient absolute)	Selected feature number	Top-3 selected features and their chemical explanations
Raman		48	Raman shifts / wavenumbers (cm^{-1}): 1264, 1804, 3112 Functional group/ vibration: υ (CC) alicyclic, aliphatic chain vibrations, υ (C=O), υ (O-H)
UV		32	UV wavelengths (nm): 794, 376, 287 Chromophore/ electronic excitation: n → π*

model is determined by L. A bigger L means better expressive power, but longer training time and higher over-fit risk. A smaller L means a simpler model, but insufficient fitting power (prone to under-fit).

In order to decide the optimal L, cross validation is performed. The dataset (160 samples × 80 features) is randomly split into a training set (80%) and validation set (20%). For different L values, the accuracies (acc) on the training set and validation set are measured. The training time (ms) is also recorded.

The above cross validation process is performed 20 times. The averaged curves are shown in Fig. 4. Several facts can be observed. 1) The training and validation accuracies increase with hyper-parameter L, as the model's fitting power is improved. 2) The accuracy increases rapidly at the (0, 10) range. Then, it slowly reaches the peak at around L = 30. 3) After the peak, the validation accuracy is stabilized, without obvious drop. This means the model doesn't become over-fit even with very large L. This shows that NNRW has a good generalization ability and is resistant to over-fitting. 4) The training/fitting time of NNRW increases almost linearly with L. This is due to the matrix operations in solving NNRW. A bigger L means larger matrices and longer computation time.

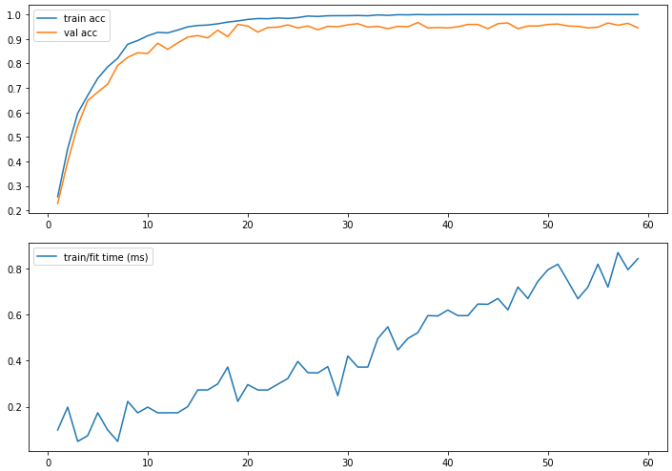


Fig. 4 The curves for ELM. The x axis means the hidden layer node number L. (1) The curve of training/validation accuracies against L. (2) The curve of training/fitting time (ms) against L. train_acc = the training accuracy. val_acc = validation accuracy. The time is measured by Python's built-in time module.

Based on the above curves, we choose L = 30 as the optimal hidden layer size. At L = 30, train_acc = 99.6%, val_acc = 96.3 %. Model fitting time = 0.372 ms.

B. Comparison with Peer Methods

The following machine learning models are analyzed and compared with NNRW. 1) MLP (multi-layer perceptron) with BP (back-propagation). We use an MLP with one hidden layer, which has the similar architecture as ELM. This MLP also has a hyper-parameter L (hidden layer nodes). 2) RBF (radial basis function)-kernel SVM (support vector machine). SVM is a widely used classification model. We use RBF-kernel SVM other than linear-SVM, as ELM is non-linear. RBF-kernel SVM has hyper-parameter $\gamma = \frac{1}{2\sigma^2}$ that controls the gaussian distribution's deviation. 3) Decision tree classifier (DTC). The decision tree has a hyper-parameter D (maximum tree depth). A bigger D gives the decision tree more expressive power to fit complex non-linear functions.

The curves of the three peer classification models are shown in Fig. 5, Fig. 6, and Fig. 7. The performance data is measured by the same cross-validation strategy on the same hardware.

For MLP (Fig. 5), the curves have similar shapes as ELM. The best cut-off point for hyper-parameter L is around 20. At L = 20, train_acc = 99.1%, val_acc = 98.0 %. Model fitting time = 57.9 ms.

For SVM (Fig. 6,), the accuracies increase with γ ($\gamma = 1/(2\sigma^2)$). A bigger γ means smaller σ (gaussian distribution deviation) and better fitting power. When γ increases, the accuracy goes up and the training time goes down. At γ = 0.0003 (σ = 41), SVM arrives at the peak accuracy. train_acc = 98.8%, val_acc = 98.4 %. Model fitting time = 2.033 ms.

For DTC (Fig. 7), its accuracy improves with D. After D reaches 4, its accuracy and training time stabilize. That means, for this dataset, DTC needs maximumly 4 levels to reach its best performance. At D = 4, train_acc = 99.6%, val_acc = 92.8 %. Model fitting time = 3.351 ms.

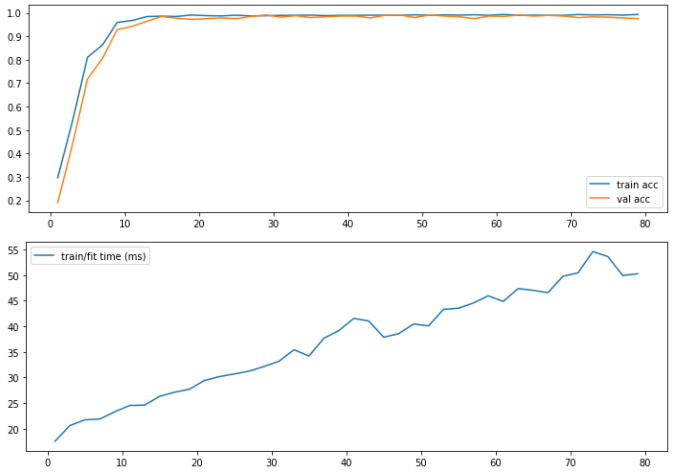
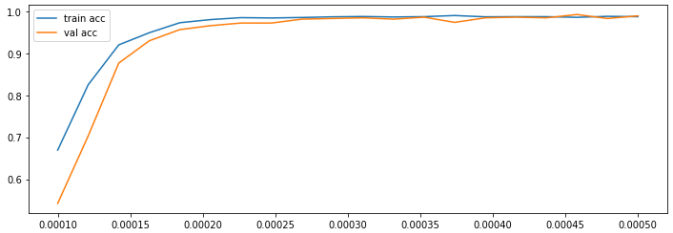


Fig. 5 The curves for MLP. The x axis means the hidden layer node number L.



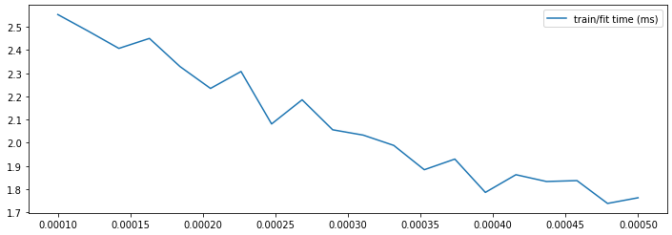


Fig. 6 The curves for RBF-kernel SVM. The x axis is the hyper-parameter $\gamma=1/(2\sigma^2)$.

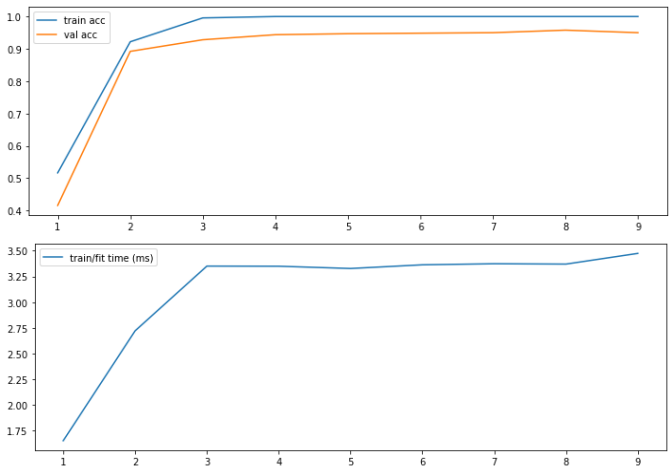


Fig. 7 The curves for decision tree. The x axis is the tree maximum depth D.

Table 2 shows the summary of different models. Several facts can be observed. 1) Considering the training and validation accuracies, all models are very close. 2) SVM (RBF-kernel) has the worst training accuracy, but the best validation accuracy. The difference between the two accuracies are the smallest. This indicates SVM has an excellent generalization power. 3) Contrary to SVM, DTC has the best training accuracy, but the worst validation accuracy. This indicates insufficient generalization power for DTC. 4) The training/fitting time of NNRW is by far the shortest. It is almost 10 times smaller than SVM and DTC, and 100 times smaller than MLP. 5) The training/fitting time of MLP is the longest. This is due to the BP training algorithm that involves gradient-based iterations.

In conclusion, NNRW shows an impressive performance and speed advantage, while maintaining a decent accuracy. It is a promising candidate for edge-side classification tasks.

Table 2 Comparison of classification models

Model	Training accuracy	Validation accuracy	Training time (ms)
NNRW	99.6%	96.3%	0.372
MLP	99.1%	98.0%	57.9
SVM	98.8%	98.4%	2.033
DTC	99.6%	92.8%	3.351
best:	**NNRW, DTC**	**SVM**	**NNRW**
worst:	**SVM**	**DTC**	**MLP**

V. Discussion

Discussions and future work related to this study are as follows.

A. Autonomous Online Learning

In data-driven and machine learning-based applications, model training is a computationally intensive task. This study shows that the neural network model with random weights (NNRW) can achieve a much better performance (only use 1/10 or 1/100 training time) than mainstream peer models, without compromising its prediction accuracy.

The NNRW-based classification method will facilitate edge-side online learning and self-training. Through online learning, the edge side can quickly evolve and adapt to the newly generated data. In this way, autonomous and intelligent agents can be achieved.

B. Ensemble Learning with NNRW

For NNRW, the accuracy and performance are two conflicting factors. We cannot improve one without sacrificing the other (Fig. 4). The balance between the two factors is modulated by the hyper-parameter L (hidden layer nodes).

In the case study, NNRW gets the best accuracy at L = 30. However, we can take a smaller L to further boost its performance/speed, at the expense of certain accuracy loss. This means NNRW becomes a "weak" but "faster" classifier, which could benefit ensemble learning scenarios. In ensemble learning, the boosting family uses multiple weak classifiers. Each weak classifier emphasizes on the misclassified samples from the previous classifier and tries to minimize the overall bias. With the extremely fast training speed, the weak version of NNRW can be used to build efficient boosting models.

C. Data Fusion with other Spectrscopic Profiling Modalities

Multi-modality is a pervasive phenomenon. The various spectroscopic profiling instruments have further extended our biological senses. Because each spectroscopic modality only portrays a specific aspect of the physical object, multi-modal data fusion gives a more comprehensive and complementary portray for the test object.

This study uses two spectroscopic profiling modalities, i.e., Raman and UV. Each one gives a different physio-chemical interpretation. As shown in Table 1, Raman spectrum reveals vibrational function groups and UV reveals molecule chromophores. Besides Raman and UV, TOF MS is also widely used. TOF MS profiles the particle *m/z* values, with more dimensions and higher resolutions. Data fusion with these modalities can further improve the model accuracy.

Supplement

The dataset, source code and case study report have been uploaded to the public repository.

DOI: 10.21227/69z3-aw19

URL: http://doi.org/10.21227/69z3-aw19

License: Creative Commons Attribution (CC-BY 4.0)

File list: 734b.csv – the data set in CSV (comma-separated value) format; case_study.pdf – the Python code and Jupyter notebook for compressed sensing; tutorial.mp4 – demo and tutorial video; src.zip – core source code of the prototype system.

A demo version of the prototype system is hosted at http://spacs.brahma.pub/CS.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (91746202, 61806177, 71433006 and 61602217), and China Scholarship Council (201808330609).

REFERENCES

[1] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society: Series B (Statistical Methodology). 67 (2005) 301–320.

[2] J. Ranstam, J.A. Cook, LASSO regression, British Journal of Surgery. 105 (2018) 1348–1348. https://doi.org/10.1002/bjs.10895.

[3] Y. Zhang, H. Wang, Building an Information Infrastructure of Spectroscopic Profiling Data for Food-Drug Quality and Safety Management, Enterprise Information Systems. (n.d.). https://doi.org/10.1080/17517575.2019.1684567.