Application of Machine Learning Models for Predicting Glucose-Level in the Pure Fluid with Algorithm for Reducing Data Dimension Based on Data Series Extraction

Tri Ngo Quang, Tung Nguyen Thanh(✉), Huong Pham Thi Viet, and Huy Bui Quang

International School, Vietnam National University, Hanoi, Vietnam
{tung_nt,huongpv}@vnu.edu.vn

Abstract. The phenomenon that glucose level of pure liquid is able to define patterns of Raman spectroscopy was demonstrated in several studies. Nevertheless, it is difficult to predict glucose level accurately by manual methods so machine learning techniques are proposed to support it. In the range of the report, we employ three simple machine learning models including Extra Trees, Random Forest, and SVM to predict glucose level from Raman spectroscopy of pure water-mixed fluid which we collected by infrastructures of Vietnam National University. In addition, the Raman data was simplified by dimension reduction algorithms based on handling data series. The results show the effectiveness of the machine learning models for predicting glucose levels as well as the reduction dimension algorithms for enhancing the performance of machine learning techniques.

Keywords: Raman spectroscopy · machine learning · Diabetes

1 Introduction

The traditional method for diagnosing diabetes is an invasive technique which causes tiny injuries as well as uncomfortable experiences to the patients. Thus, several solutions tend to predict glucose level without drawing blood of patients by implanted lancet and some of them apply the close relation between glucose level and Raman spectroscopy from the reflection in human skin. Numerous studies showed the high viability of non-invasive blood glucose monitoring by Raman waveshift, but the main problem was the challenging data collecting and insufficient precision in calculating blood glucose levels from Raman spectroscopy data. We build on earlier research to suggest a method for predicting glucose level from Raman data in the context of artificial intelligence using machine learning techniques. As a result, we view the measurement as a classification issue involving different glucose-level labels. Our objective in the context of this study is to evaluate the effects of various data pre-processing techniques and validate the potential for glucose-level prediction from Raman waveshifts using machine learning techniques. We have gathered standard data with a high degree of purity and little noise by measuring

P. Cong Vinh and N. Thanh Tung (Eds.): ICCASA 2023, LNICST 579, pp. 87–101, 2024.
https://doi.org/10.1007/978-3-031-58878-5_6

the Raman spectrum of deionized water with a clear glucose level. Additionally, the moderate complexity and broad coverage of the dataset we have gathered make it ideal for the validation objectives of fundamental machine learning models. In order to reduce the complexity of Raman datasets, we also suggest a novel preprocessing technique. In this study, we examined the data's format and put into practice a hotspot series-based data reduction technique. The outcome is very encouraging and shows how machine learning can be used to predict glucose levels using Raman waveshift in addition to showing the benefits of various data pre-processing techniques.

The paper consists of four major sections: theoretical background, technique for analyzing and pre-processing Raman datasets, experiments, and conclusion.

2 Theoretical Background

In this chapter, we describe some general theoretical findings about our project, including non-invasive glucose-level measurement using the Raman spectroscopy and machine learning.

2.1 Non-invasive Glucose Measurement Using Raman Spectroscopy

Raman measurement used the Raman effect, which was discovered by Raman and Krishnan in 1928 [1]. Raman spectroscopy, based on this effect, is a scattering technique. When a sample is exposed to monochromatic laser sources, the molecules in the sample interact with the lasers and scatter light. A Raman spectrum is produced by scattering light at a frequency different from inelastic scattering. The inelastic interaction of sample molecules with monochromatic light generates the Raman spectrum. The measuring instrument that uses Raman spectroscopy is known as a Raman spectrometer and consists of four modules: a laser generator, a chamber for storing measuring samples, a grating-equipped spectrometer chamber, and a detecting system [2].

Before the use of Raman spectroscopy in healthcare was found to be effective, the measurement acquired by non-invasive methods was of concern. For example, Caduff et al. proposed a non-invasive method to predict glucose level from impedance spectroscopy [18]. With the relationship between Raman data and glucose-level, there were several studies that raised huge concerns in the scientific community. Xu et al. [3] described available targets of measurement using Raman spectroscopy as a large amount of substances. Thus, the relationship between Raman spectroscopy reflected from human bodies and blood glucose-level was discussed in [4]. According to this study, blood glucose levels in living animal bodies can be determined in vivo using the Raman spectra obtained from a diode-laser operating at 785 nm. By using partial least squares techniques, the glucose-level of humans can be measured using near-infrared Raman spectroscopy [17]. The research in [19] had a similar approach, but the statistical techniques were classical least squares, principal component regression and partial least squares.

These investigations demonstrate that it is entirely feasible to forecast the glucose level using Raman spectroscopy. Due to the significant difficulty in observing the relationship between Raman data and glucose level, the problem we had to solve was the

selection of a prediction technique. We may also notice that several statistical methods have demonstrated their effectiveness in identifying characteristics between Raman spectroscopy and glucose levels. According to previous research and our own findings, the machine learning technique is just as effective as the methods mentioned above. The possibility and approach of machine learning algorithms for glucose prediction based on Raman spectroscopy were addressed in the following section.

2.2 The Use of Machine Learning for Predicting Glucose-Level from Raman Spectra

The Potential of Machine Learning in Glucose Concentration Prediction. Machine learning has proved its efficiency in problems of prediction and classification [9]. There were several former studies about the application of machine learning on glucose-level classification from extracting Raman data. Some of them concentrated on detecting diabetes from this kind of data. The research in [10] proposed a non-invasive glucose measurement using 5 simple machine learning algorithms from optical sensors. In the study, the dataset was the 18 pairs which each pair had 2 values: wavelength and intensity. The techniques for machine learning included a Feedforward Neural Network for multi-class classification. In this investigation, we investigated the experimental methodology employed by the authors when they collected glucose-distilled water samples to simulate human blood. The objective of this method is to generate a simple dataset with a high degree of accuracy and a predicted value.

In article [16], the author collected Raman spectroscopy data on blood samples from normal people and diabetes patients before proposing an algorithm based on a combination between Principal Component Analysis and Linear Discriminant Analysis to classify [16]. The advantage of the study, when compared to [10] is that Raman spectroscopy was the sole input data to be classified by the algorithms. As reported by [21], the results and classification discretization framework of an experiment using a visible near-infrared laser derived from an optical sensor to measure glucose level were presented and shown to be promising. However, Raman spectroscopy was performed on human blood, which had a high degree of accuracy despite the intrusive nature of blood collection. In addition, 76 individuals were sampled, including 39 diabetes patients and 37 healthy individuals. The small quantity of this number reduces its persuasiveness. The research in [11] introduced a portable spectrometer for non-invasive glucose testing based on the Raman effect. Positive aspect of the study is the in vivo studies conducted on diabetes patients and healthy individuals. Using the spectrometer, the solution acquired Raman spectroscopy from certain regions of the human body via a non-invasive measurement. Support Vector Machine and Artificial Neural Network were used as machine learning prediction models. Some other benefits of the study were the data processing techniques that enhanced the performance of machine learning algorithms. The first problem with this study is the small sample size, which consists of just about 20 individuals. The second categorization is lean, for which it is hard to specify a precise glucose level. Instead, the answer predicted whether or not a sample was obtained from a diabetic patient.

We acquired a number of outstanding perspectives and approaches from the aforementioned studies, while avoiding their flaws. We continue to reference these research

while focusing on the outcome and measuring metric. In this section, we present simply the approach and theoretical foundation.

Applied Machine Learning Model.
In the scope of study, we use 3 machine learning models for classification, including Extra Trees, Random Forest and Support Vector Machine (SVM):

- The Extra Trees model:

The Extra Trees machine learning model was proposed based on tree architecture [5]. The Extra-Trees approach follows the traditional top-down method and employs a collection of unpruned decision or regression trees. The number of characteristics that are randomly chosen at each node and the minimal sample size for splitting a node are the two parameters for the Extra-Trees splitting technique for numerical attributes. It is applied several times to the whole original learning sample in order to produce an ensemble model, which we identify by the number of trees in the ensemble. By majority vote in classification problems and arithmetic average in regression problems, the predictions of the trees are combined to get the final prediction.

The Extra Trees model has many applications in detection and classification. A solution for detecting phishing websites integrated Extra Trees algorithms and AI Meta-Learners techniques have been proposed in [6]. Another application of Extra Trees is the Autotrophic/Heterotrophic Microorganism Mixtures using absorbance spectrum data [7].

- The Random Forest model:

The Random Forest model was created by integrating many tree-based data structures and randomizing the dataset and was found to be an effective classifier for bio-sensor data [8]. A number of tree classifiers were integrated with Random Forest based on combination regulation. Each of these classifiers casts a vote for the class with the most members, and the final sort result is produced by combining these votes. This algorithm is distinguished by its high classification precision, tolerance for noise and outliers, and lack of overfitting. During the creation of Random Forest, the tree is also planted on the new training set using random feature selection, and the new training set is taken from the previous training set using bagging methods.

Similar to other methods for machine learning, Random Forest can be used for classification and regression. In [12], several uses of Random Forest for managing complex data from remote sensors are discussed. From the list of these applications, we determined that Random Forest was capable of dealing with a number of data sources, such as multi-spectral radar, sensing images, and hyperspectral imagery. Using Raman spectroscopic data and Random Forests has another use: locating substances used in nanofabrication [13]. The computationally feasible analysis of genome-wide association data using Random Forest is one of the first instances presented in this work.

- The Support Vector Machine model:

The Support Vector Machine (SVM) has a huge variety of supervised applications with different data sources [14]. The capacity of the SVM to learn data classification

patterns with a balance between accuracy and reproducibility is what gives it its power. It has gained popularity as a classification tool, though it is still infrequently employed for regression tasks. It is highly versatile and may be utilized in a variety of data science contexts, including the study of brain illnesses. In order for the SVM to function, a hyperplane that optimizes the separation between the support vectors of the two class labels was selected. In comparison to other kinds of classifiers, the SVM's capability and attraction stem mostly from its ability to give balanced performance even when the complexity of the feature space greatly exceeds the number of training data. In addition, the SVM provides diversity. For the SVM decision functions, many distinct kernel functions can be provided, and most software enables users to choose unique kernels. This functionality makes it easier to employ the SVM classifiers to solve linear classification problems without having to spend a lot of time on hyperparameter adjustment. The SVM is effective in solving a variety of classification issues with high dimensional data [20]. Therefore, it can be applied efficiently to Raman spectra datasets.

Both three machine learning models, including Extra Trees, Random Forest and SVM were fully supported in Sci-kit Learn – a Python library for ML implementation. Thus, we use this for developing our project.

3 Collection and Dimensions Reduction of Raman Dataset

3.1 Collection of Dataset

To evaluate the accuracy of machine learning models in predicting glucose levels for the objectives of this study, sample-level precision was crucial. According to prior study, there are three drawbacks to the samples collected through non-invasive or invasive measures in regions of the human body [10]. The dense appearance of noise, the difficulty of obtaining a sample with the desired value, and the medical ethics surrounding human experimentation. We produce a pseudo-sample with the desired value based on these challenges.

With the purpose of replicating blood samples with a determined glucose level in our laboratory, we created samples by combining pure glucose and deionized water in a particular ratio. Main components used for measurements consist of:

- Raman spectrophotometer: uRaman - Ci, Technospex.
- Glucose chemical products: 99.5% Sigma-Aldrich glucose.
- Deionized water solution bottle.
- Tools (solution tube, stirring rod,…).

The measurement procedure is described as below:

1. Set excitation laser power to 100 mW.
2. Set the measurement range at the wavelength of 785.1 nm in 300 s.
3. Place 3 ml of glucose solution from each of the mentioned test tubes on the quartz surface of the Cuvette.
4. Insert the cuvette (with the MACRO-CH/Quartz Cuvette accessory) into the measurement chamber of the uRaman – Ci spectrophotometer.
5. Collect digitized Raman signals from the measurement experiment.

The set of devices used to acquire Raman spectroscopic data by analyzing samples of glucose-mixed fluids with Technospex uRaman - Ci. We have not yet adopted non-invasive measurement in human tissue due to the side effects of external variables in the Raman data from this measurement were too significant to permit an evaluation of the correlation between glucose level and sample figures.

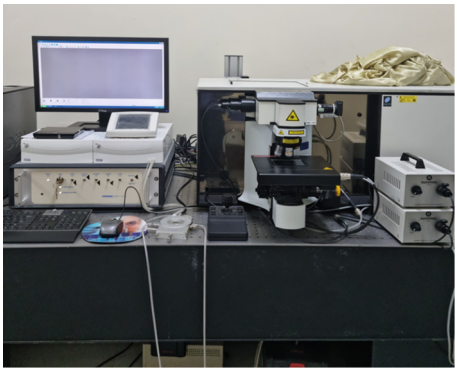

Fig. 1. Device set used for Raman spectroscopy measurement from artificial glucose-mixed fluids.

Within the scope of the study, our dataset contains 50 samples with 50 labels, but only 10 label-values. The Raman shift data is contained in a CSV file, and its label is the prefix of this CSV file, which is separated from the other portions by a "-" character. Each label in the set indicates a glucose value and was encoded as an integer: I = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. The encoding process was handled by a function which analyzed the CSV file. Basic description about our dataset is described in Table 1:

Table 1 Description of the dataset used for the machine learning algorithm**.**

Glucose concentration level (mmol/L)	Number of samples	Label code
5.0	5	1
5.5	5	2
6.0	5	3
6.5	5	4
7.0	5	5
7.5	5	6
10.0	5	7
12.5	5	8
15.0	5	9
20.0	5	10

The distance between values indicated by labels is diverse, as seen in Table 1. Between 15.0 and 20.0, the shortest distance is 0.5 m and the longest is 5.0 m. The gap between glucose levels is not considered as an extraction criterion because the machine learning technique was built to handle only classification problems. Yet, under conventional settings for measuring blood samples, glucose concentration is the most influential element in determining the result pattern. If the difference in glucose levels between two samples was modest, there would be a high degree of data pattern similarity, which would make machine learning algorithm extraction more challenging. Consequently, the dataset enables us to test the performance of several machine learning models for extracting features from different glucose feature distances.

3.2 The Hotspot Series Extraction Procedure

The Primitive Input Data.
Additionally, the moderate complexity and broad coverage of the dataset we have gathered make it ideal for the validation objectives of fundamental machine learning models. In order to reduce the complexity of Raman datasets, we also suggest a novel preprocessing technique. In this study, we examined the data's format and put into practice a hotspot series-based data reduction technique. The outcome is very encouraging and shows how machine learning can be used to predict glucose levels using Raman waveshift in addition to showing the benefits of various data pre-processing techniques. Based on this explanation, we define each sample's input data as an array of intensity values, where each intensity value's index corresponds to its index within the sample. The fundamental input data for the machine learning method is specified as a 2048-element array of intensity values sorted ascendingly by matching waveshift throughout the scope of the study.

Hotspot Segments of Primitive Data.
We recognize that the primitive input data of a sample is extremely complicated and needs to be reduced. In each labeled-group, we randomly selected one of five samples and plotted it using the Python tool matplotlib, as shown in Fig. 2.

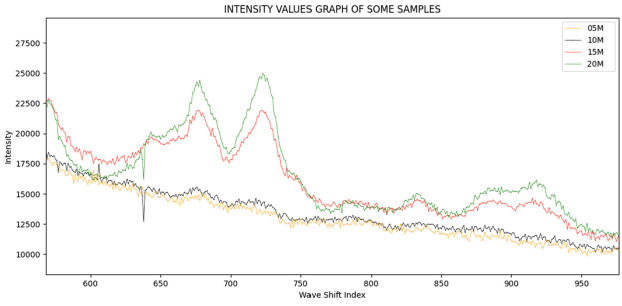

Fig. 2a Large distance between glucose levels of samples: 5, 10, 15 and 20

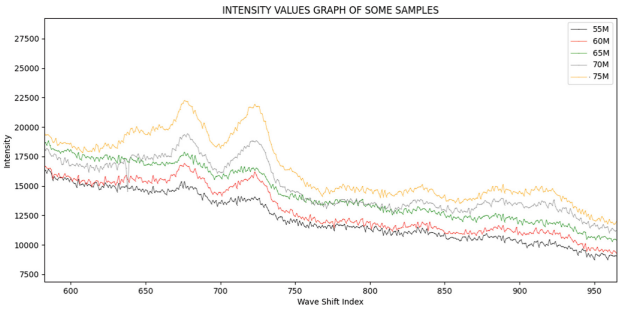

Fig. 2b Small distance between glucose levels of samples: 5.5, 6.0, 6.5, 7.0 and 7.5 (The graph of intensity values in some samples with different glucose level)

Figure 2 contains two plots with distinct glucose level separations. In Fig. 1a, the distance is 5, and the range of glucose levels is from 5 to 20. In contrast, the distance in Fig. 1b is only 0.5, and the range of glucose levels is between 5.5 and 7.5. We can observe that:

- In all samples, the ascending sequence of wave shift does not provide a persistent change in intensity. There are a considerable number of 10-value curves with the maximums in the initial 1250 indexes. Throughout the last 750 indices, the number has fluctuated modestly.
- In the first indexes of these graphs, the difference between samples' glucose concentrations is expressed clearly. In the range of indexes from 500 to 1250, visibility is excellent.

This finding leads us to the conclusion that the length of input data is enormous, but not all values provide useful classification information for machine learning models. Hence, we define a hotspot as a chunk of input data containing useful categorization features. By examining the values of the smaller-than-input-data hotspot region, machine learning models may collect nearly all the characteristics of a sample with a certain glucose level. Although we can define hotspot segments through observation, we build a method to detect them with greater precision.

Generation of Intensity Series.
First, we define an intensity series as a portion of input data containing a predetermined number of sequence values. As mentioned previously, each sample's input data is an intensity array. This array can be represented in the following mathematical formula (1):

$$A_i(g) = \{I_j \in N | j \in [0, 2047]\} \tag{1}$$

In the formula (1):

- $A_i(g)$: array of intensity values with index i and glucose level g
- I_j: intensity value with index j

This array is used to construct an intensity series of length l (l < 2048) by selecting l sequence elements. Hence, the distinct series are produced by assigning distinct elements

to the series' initial constituents. The generic formula (2) provides the mathematical structure of a length l intensity series:

$$S_j(A_i(g)) = \{I_k \in N | l \in [1, 2047],$$

$$j \in [0, 2048 - l], k \in [j, j + l]\} \tag{2}$$

In the formula (2):

- j: starting index of an intensity series
- $A_i(g)$: array of intensity values with index i and glucose level g
- $S_j(A_i(g))$: an intensity series of $A_i(g)$ with starting index j
- I_k: intensity value with index k. This index is setted in $A_i(g)$.
- l: the size of the intensity series

Because the intensity array has 2048 indexes, each sample has a set of (2049 - l) series with a separate starting index. From this set, we select the hotspot series and generate a new array that contains all elements of the hotspot series.

Extraction of Hotspot Series.
After extracting the intensity series, the next step is selecting the hotspot series and defining the new input data with similar extraction of all samples. Thus, the extraction process includes 3 stages: calculate hotspot level, select the hotspot series based on the hotspot level and output new input data for the machine learning models.

In the first stage, we calculate hotspot level based on median and average variance of this series in the Formula (3) and (4):

$$\underline{S}_j(A_i(g)) = \frac{1}{l} \sum_{j+l}^{j} \quad (I_j) \tag{3}$$

In the formula (3):

- $A_i(g)$: array of intensity values with index i and glucose level g
- $\underline{S}_j(A_i(g))$: a median of the intensity series of $A_i(g)$ with starting index j
- I_j: intensity value with index j
- l: the size of the intensity series

These is formula (4)

$$V_j(A_i(g)) = \frac{1}{l} \sum_{j+l}^{j} \quad \left(I_j - \underline{S}_j(A_i(g))\right)^2 \tag{4}$$

In the formula (4):

- $A_i(g)$: array of intensity values with index i and glucose level g
- $V_j(A_i(g))$: average variance of the intensity series of $A_i(g)$ with starting index j
- $\underline{S}_j(A_i(g))$: a median of the intensity series of $A_i(g)$ with starting index j
- I_j: intensity value with index j
- l: the size of the intensity series

After calculating the average variance of each array's series, we chose a set number of the series with the highest average variance. Then, new array input data are defined by picking all indexes of the intensity array that are contained in the selected series of all samples. The new input data is the intensity array from which all intensity values whose indices are not selected have been removed.

4 Experiment

4.1 Setup of Experiments

We implemented our machine learning algorithm with three models, including Extra Trees, Random Forest and Support Vector Machine on a personal computer with adequate software and hardware configuration. Our computer has 8GB of RAM capability and a Microsoft 64-bit operating system. Meanwhile, Python was selected to implement this algorithm because this programming language has a large number of libraries that strongly support machine learning models. Specifically, we use Python-based libraries such as Scikit-learn to construct these machine learning models as well as Mathplot-lib to write graph about Raman spectra in Fig. 2.

4.2 Creation of Experimental Dataset

Implementation of the Hotspot Series Extraction Algorithm.
We implement the hotspot series extraction algorithm using Python. We set the size of the series as 10 values and the number of selected series in each sample is 500 series. With this dataset, the result is that 831 values are selected with some segments including index from 0 to 29, index from 31 to 542, index from 553 to 586, index from 629 to 647, index from 664 to 699, index from 703 to 743, index from 782 to 799, index from 992 to 1010, index from 1109 to 1122, index from 1171 to 1207, index from 1505 to 1523, index from 1620 to 1642, index from 1706 to 1715 and index from 1865 to 1883. In conclusion, the new input data has 831 values which contain 40.5% in comparison with the size of primitive data.

The next step is to divide this experimental dataset into a train set and a test set. We subdivided the dataset based on k-fold validation to include data for training and testing.

Separation of Dataset.
The next step is to divide this experimental dataset into a train set and a test set. We subdivided the dataset based on k-fold validation to include data for training and testing.

To avoid overfitting, we implement k-fold cross validation, which is the process of using each subset as the test data set and the remaining subsets as the training data. It involves breaking a data set into k subsets. The performance metrics for each validation process are then averaged. There is not a single best indicator for evaluating machine learning algorithms because each approach has advantages and disadvantages. In the experiment, we divided the population equally into 5 subsets, with one subset used for the test set and the other four subsets used for the training set.

The test set for root dataset consists of 10 samples, each of which contains a distinct label-value from a collection of 10 label-codes. Currently, the train value data consists

of forty samples, four of which have identical label-codes. The test set comprises six samples, each of which has a unique label-value selected from a collection of six label-codes. In the interim, the train value data includes 24 samples, four of which have identical label codes.

In addition, there are 450 iterations of training with each machine learning model including Extra Trees, Random Forest and SVM models.

4.3 Measuring Criterions

The algorithm's efficiency is determined by the accuracy of the classification process with the data from the test set. Specifically, the algorithm with a specific machine learning model classifies a sample regardless of its label-code and defines a label-code for this sample. After that, the algorithm compares the predicted label-code to the existing label-code of this sample. There are possible 2 cases of this comparison:

- *True (T):* The predicted label-code and existing label-code is the same.
- *False (F):* The predicted label-code and existing label-code is different.

There are 4 criterions used to determine the effectiveness of our model including: Accuracy (Acc), Specificity (Sp), Sensitivity (Se) and ROC-AUC of our model. Meanwhile, Specitivity, Sensitivity and ROC-AUC are calculated by One-vs-rest (OvR) strategy:

The accuracy of our model is defined by Formula (5):

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}, \tag{5}$$

The model's Specificity (Sp) is defined as:

$$Sp = \frac{TN}{TN + FP} \tag{6}$$

The Sensitivity (Se) is defined as.

$$Se = \frac{TP}{TP + FN}, \tag{7}$$

where

- TP – Number of true positive instances
- TN – Number of true negative instances
- FP – Number of false positive instances
- FN – Number of falsenegative instances

Each machine learning model consists of $i = 450$ loops in which subsets are differently partitioned into train and test sets prior to training and accuracy values are determined after each turn. Afterwards, the mean metrics accumulated over 450 iterations are calculated. There are 3 ML model used in our scope of investigation: Extra Trees, Random Forest, and Support Vector Machine; thus, there are three values of average Accuracy for each model.

4.4 Results of Experiments

This section discusses the outcome of applying our selected augmentation strategies for Raman spectroscopy to several ML models. The following are 3 scenarios for the dataset:

- Root dataset - without using any preprocessing method
- Hotspot series dataset – Using hotspot series extraction applied to the root dataset.

Root Dataset - Without using Hotspot Series Extraction.
Table 2 shows the result with each model in case of primitive data without any conversion. In each sample, the number of intensity figures is 2048. The metrics are described in percentage unit (%):

Table 2 Result of experiment with each model in case of root dataset.

	Mean result over 450 iterations			
	Acc	Sp	Se	ROC-AUC
Extra Trees	92.64	99.19	92.73	99.23
Random Forest	87.24	87.47	85.40	98.59
Support Vector Machine	84.43	98.38	86.21	99.04

The Extra Trees model has the highest accuracy score in this table, whereas the Random Forest and SVM models have lower Average Accuracy values. The accuracy ratings range from 84% to 92%, which is a moderately good range. Similar to the accuracy scores, the Extra Trees model has greater Specificity, Sensitivity, and ROC-AUC values than the other models. While the SVM model's estimate of its learning capacity is superior, its performance with this data is the worst. Yet, when comparing one component of One-vs.-Rest Specificity to another, we detect an anomalous characteristic. The SVM model has a better specificity than Random Forest, which is distinct from the three other measures. It indicates that the SVM model is more resistant to type I errors due to its unique processes for tackling classification issues.

Preprocessed Data - Using Hotspot Series Extraction Algorithm.
Table 4 shows the result with each model in case of data after using hotspot series extraction. In each sample, the number of intensity figures is only 831 which means its size is 40.5% of the size of the root dataset. The basic unit is also percent (%):

Because accuracy is the most crucial criterion, we compare value between different models as well as between root and preprocessed dataset and describe visually in the Fig. 3:

From Table 3 and Fig. 6, the Extra Trees model has the highest accuracy score in this table, whereas the Random Forest and SVM models have lower Average Accuracy values. The accuracy ratings range from 84% to 92%, which is a somewhat good range. Similar to the accuracy scores, the Extra Trees model has greater Specificity, Sensitivity,

Table 3 Result of experiment with each model in case of hotspot series preprocessed dataset

	Mean result over 450 iterations			
	Acc	Sp	Se	ROC-AUC
Extra Trees	93.77	99.30	93.67	99.48
Random Forest	88.24	98.70	99.30	99.02
Support Vector Machine	89.63	98.84	98.60	99.37

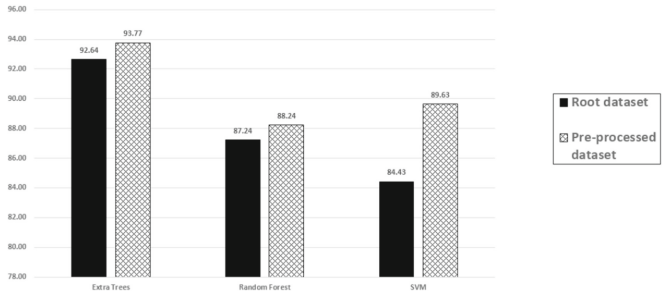

Fig. 3: The chart of values in some accuracies with different kind of input data and machine learning models

and ROC-AUC values than the other models. While the SVM model's estimate of its learning capacity is superior, its performance with this data is the worst. Yet, when comparing one component of One-vs.-Rest Specificity to another, we detect an anomalous characteristic. The SVM model has a better specificity than Random Forest, which is distinct from the three other measures. It indicates that the SVM model is more resistant to type I errors due to its unique processes for tackling classification issues.

5 Conclusion

Based on studies into the correlation between a person's glucose level and Raman spectroscopy reflected from various bodily areas, we suggested a method for assessing glucose utilizing a Raman spectrometer and a machine learning system with many models. Before the machine learning models anticipate the glucose level from the samples, the spectrometer generates Raman spectra from human samples. We utilized the Technospex uRaman - Ci spectrometer to build a dataset from glucose-mixed fluids with varying glucose concentrations. Before being labeled as primitive data, the dataset has been cleansed of noise. We also utilized feature-extraction techniques to enhance the performance of machine learning systems by reducing the size of input data. The preprocess data based on simple calculation to define data series bring valuable information. Thus, the preprocess data becomes shorter, but brings features of glucose levels. The input data for the machine learning model could be basic data or data extracted during the extraction procedure.

Before testing the accuracy of the trained models' classification of samples, we designed experiments in which machine learning algorithms extracted characteristics from the dataset. Experiments utilized three models: Extra Trees, Random Forest, and SVM model. The results proved the efficiency of machine learning on classification problems as well as the preprocessing procedures that meet our requirements. The accuracy ranges from 80% to 97%, while the extraction process increases the accuracy of each machine learning model in the same experimental dataset.

We will raise the dataset's complexity in the future by increasing the number of labels and the number of samples for each label. Non-invasive sampling methods for Raman spectroscopy data collection will also be thoroughly investigated, and we will continue to improve the machine learning models and preprocessing techniques used to extract characteristics from the dataset.

References

1. Raman, C.V., Krishnan, K.S.: A new type of secondary radiation. Nature **121**(3048), 501–502, (1928)
2. Schmid, T., Dariz, P.: Raman microspectroscopic imaging of binder remnants in historical mortars reveals processing conditions. Heritage **2**(2), 1662–1683 (2019)
3. Jun, X., et al.: Raman spectroscopy as a versatile tool for investigating thermochemical processing of coal, biomass, and wastes: recent advances and future perspectives. Energy Fuels **35**(4), 2870–2913 (2020)
4. Todaro, B., et al.: "Is Raman the best strategy towards the development of non-invasive continuous glucose monitoring devices for diabetes management?." Front. Chem.**10**, 994272 (2022) https://doi.org/10.3389/fchem.2022.994272
5. Yang, S.J., et al.: "Rapid identification of microplastic using portable Raman system and extra trees algorithm." Real-time Photonic Measurements, Data Management, and Processing V, Vol. 11555. SPIE, 2020
6. Alsariera, Y.A., Adeyemo, V.E., Balogun, A.O., Alazzawi, A.K.: AI meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access **8**, 142532–142542 (2020)
7. Nakanishi, A., et al.: "Development of a Prediction Method of Cell Density in Autotrophic/Heterotrophic Microorganism Mixtures by Machine Learning Using Absorbance Spectrum Data." BioTech **11**(4), 46 (2022):
8. Sadat-Mohammadi, M., et al.: "Non-invasive physical demand assessment using wearable respiration sensor and random forest classifier." J.Build. Eng. **44**, 103279 (2021)
9. Khan, Z.Y., Niu, Z., Sandiwarno, S.: Deep learning techniques for rating prediction: a survey of the state-of-the-art. Artif. Intell. Rev. **54**, 95–135 (2021)
10. Shokrekhodaei, M., Cistola, D.P., Roberts, R.C., Quinones, S.: Non-invasive glucose monitoring using optical sensor and machine learning techniques for diabetes applications. HHS Public Access, IEEE Access **9**, 73029–73045 (2021)
11. Guevara, E., Torres-Galván, J.C., Ramírez-Elías, M.G., Luevano-Contreras, C., González, F.J.: Use of raman spectroscopy to screen diabetes mellitus with machine learning tools. Biomed. Opt. Express, 9(10): 4998–5010, 2018
12. Belgiu, M., Dragut, L.: Random forest in remote sensing: a review of applications and future directions. ISPRS J. Photogramm. Remote Sens. **114**, 24–31 (2016)
13. Theobald, N., et al. "Identification of unknown nanofabrication chemicals using raman spectroscopy and deep learning." IEEE Sens. J. (2023)

14. Pisner, D.A., Schnyer, D.M.: Support vector machine. Machine Learning, Chapter 6: 101–121, Academic Press (2020)
15. Sujay Raghavendra, N., Deka, P.C.: Support vector machine applications in the field of hydrology: a review. Appl. Soft Comput. **19**, 372–386 (2014)
16. Lin, J., et al.: Raman spectroscopy of human hemoglobin for diabetes detection. J. Innovative Opt. Health Sci. **7**(1), 1350051–1350056 (2014)
17. Berger, A.J., Itzkan, I., Feld, M.S.: Feasibility of measuring blood glucose concentration by near-infrared Raman spectroscopy. Spectrochim. Acta Part A Mol. Biomol. Spectrosc. **53**(2), 287–292 (1997)
18. Caduff, A., Hirt, E., Feldman, Y., Ali, Z., Heinemann, L.: First human experiments with a novel non-invasive, non-optical continuous glucose monitoring system. Biosens. Bioelectron. **19**(3), 209–217 (2003)
19. Ehsan, U., et al.: "Surface-enhanced Raman spectroscopy of centrifuged blood serum samples of diabetic type II patients by using 50KDa filter devices." Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy **293**, 122457 (2023):
20. Zoppis, I., Mauri, G., Dondi, R.:"Kernel methods: Support vector machines." Encyclopedia of Bioinformatics and Computational Biology. Volume 1. Elsevier, 503–510 (2019)
21. Shokrekhodaei, M., et al.: "Non-invasive glucose monitoring using optical sensor and machine learning techniques for diabetes applications." IEEE Access **9**, 73029–73045 (2021)