EDINBURGH INSTRUMENTS
RMS 1000
RAMAN MICROSCOPE
EDINBURGH INSTRUMENTS
RM5
RAMAN MICROSCOPE

Received: 29 April 2022 | Revised: 14 July 2022 | Accepted: 9 August 2022

DOI: 10.1002/jrs.6447

Journal of RAMAN SPECTROSCOPY WILEY

RESEARCH ARTICLE

Machine learning using convolutional neural networks for SERS analysis of biomarkers in medical diagnostics

Joy Qiaoyi Li[1,2] | Priya Vohra Dukes[4] | Walter Lee[4,5] | Michael Sarkis[6] | Tuan Vo-Dinh[1,2,3]

[1]Fitzpatrick Institute for Photonics, Duke University, Durham, North Carolina, USA

[2]Biomedical Engineering Department, Duke University, Durham, North Carolina, USA

[3]Chemistry Department, Duke University, Durham, North Carolina, USA

[4]Department of Head and Neck Surgery and Communication Sciences, Duke University School of Medicine, Durham, North Carolina, USA

[5]Global Health Institute, Duke University, Durham, North Carolina, USA

[6]Department of Statistical Science, Duke University, Durham, North Carolina, USA

Correspondence
Tuan Vo-Dinh, Fitzpatrick Institute for Photonics, Duke University, Durham, NC, USA.
Email: tuan.vodinh@duke.edu

Funding information
National Institute of Health (NIGMS), Grant/Award Numbers: 1R01DE030455-01A1, R01GM135486; U.S. Department of Defense, Grant/Award Number: NDSEG Fellow ID: 00007902

Abstract

Surface-enhanced Raman spectroscopy (SERS) has wide diagnostic applications because of narrow spectral features that allow multiplexed analysis. Machine learning (ML) has been used for non-dye-labeled SERS spectra but has not been applied to SERS dye-labeled materials with known spectral shapes. Here, we compare the performances of spectral decomposition, support vector regression, random forest regression, partial least squares regression, and convolutional neural network (CNN) for SERS "spectral unmixing" from a multiplexed mixture of 7 SERS-active "nanorattles" loaded with different dyes for mRNA biomarker detection. We showed that CNN most accurately determined relative contributions of each distinct dye-loaded nanorattle. CNN and comparative models were then used to analyze SERS spectra from a single-plexed, point-of-care assay detecting an mRNA biomarker for head and neck cancer in 20 samples. The CNN, trained on simulated multiplexed data, determined the correct dye contributions from the singleplex assay with $\text{RMSE}_{\text{label}} = 6.42 \times 10^{-2}$. These results demonstrate the potential of CNN-based ML to advance SERS-based diagnostics.

1 | INTRODUCTION

Raman spectroscopy, particularly surface-enhanced Raman scattering (SERS), has widespread applications in chemical,[1] biological sensing,[2–4] and imaging.[5] The advantages of SERS include high sensitivity and very narrow Raman peaks, enabling single molecule detection[6] and high degrees of multiplexing, respectively. These narrow peaks are an important improvement over common fluorescence techniques, which have broad emission spectra without distinct maximums, precluding multiplex analysis. These unique characteristics make the development of highly multiplexed SERS-based technology an appealing target for the advancement of molecular diagnostics.

Molecular diagnostics are powerful medical tools for the diagnosis and monitoring of a variety of diseases, including cancer,[7] infectious disease,[8] and molecular metabolic disease,[9] because of the tests' high sensitivities and specificities. The gold standard for nucleic acid molecular diagnostic tests is quantitative polymerase chain reaction (qPCR). Despite its superior sensitivity, qPCR requires access to a standard laboratory and is labor/time intensive. Additionally, qPCR utilizes target amplification, which can be prone to false positives. In limited-resource settings, where qPCR technologies are often unavailable, development of molecular diagnostic techniques has potential to revolutionize the medical field by allowing for timely diagnosis and treatment, thus improving prognosis. The high specificity, sensitivity, and multiplexing ability of SERS techniques, along with new developments in technology miniaturization, make them suitable for the implementation in point-of-care (POC) molecular diagnostics.[10]

Our laboratory has developed SERS nanosystems for a wide variety of applications spanning chemical monitoring, biomedical sensing, and theranostics.[4,5,11–15] Recently, we created a multifunctional, bimetallic, SERS nanoplatform called ultrabright SERS "nanorattles."[16,17] These nanorattles are capable of nucleic acid molecular biosensing, infectious disease detection,[16] and *in vivo* SERS sensing and imaging.[17] These particles are "ultrabright" because of their high SERS enhancement from their core-gap-shell architecture and loading with fluorescent dyes resonant at the excitation wavelength, resulting in surface-enhanced resonance Raman scattering (SERRS), further increasing SERS enhancement. Nanorattles have been used for the direct detection of malaria DNA in blood lysate[18] and the detection of a squamous cell carcinoma mRNA biomarker in unamplified RNA extracts from head and neck cancer tissue.[19,20] The nanorattles can be loaded with distinct Raman-active dyes for multiplexing. Here, we used the SERS spectra of nanorattles loaded with seven different dyes as the model system for machine learning (ML) analysis; this technique can be utilized for in vivo imaging and further developed for biosensing.

Because of the unique, narrow SERS peaks, conventional methods of signal separation have identified characteristic peak(s) for each multiplexed target.[21] This technique is highly sensitive to noise and limits the number of possible multiplexed entities, which is challenging when many established Raman-active dyes have similar structures and thus similar SERS spectra. Lutz et al. developed a spectral decomposition (SD) procedure that deconvolves multiplexed spectra by finding the best fit of normalized spectra to component reference spectra, with a free-fitting polynomial representing the background.[22] This method does not require background subtraction, eliminating a major source of error but is more sensitive to noise, leading to lower sensitivity. Principal component analysis (PCA), which performs well with noisy signals, along with its regression counterpart, principal component regression (PCR), have been used to analyze SERS spectra.[23–26] Although these dimensionality reduction techniques manage noise well, interpretation of results and quantification of individual analytes is difficult because principal components do not necessarily correspond with dye label spectra. PCA is an unsupervised technique requiring no a priori knowledge and is commonly used for non-labeled SERS applications.[27,28] Partial least squares (PLS), the supervised counterpart to PCA, can improve model predictions when labels are known in the training set. Further, partial least squares regression (PLSR) employs PLS for regression tasks. Because dye-labeled SERS multiplexing uses a set of known reference dyes, supervised techniques including PLSR can improve analysis performance.[26]

Despite the robustness of these statistical tools, multiplexed assay analyses grow more complex with increased multiplexing. Thus, there is a need for ML methods such as support vector machine (SVM),[25,29–36] decision trees such as random forest (RF)[25,36–40], and k-nearest neighbors (k-NN).[26,41,42] Recently, deep learning or multilayered ML networks, which can extract high-level features from an input, has also been applied to analyze Raman spectra. Convolutional neural networks (CNNs) are a type of deep learning using convolutional layers to learn kernel weights and extract differentiating features from an input. CNNs are classically applied to image processing tasks but have more recently been applied to analyze 1D spectra, including Raman spectra. So far, CNNs have been mostly utilized to analyze native SERS spectra of non-dye-labeled samples, including those obtained for chemical sensing,[43] cancer differentiation,[44,45] and pathogen sensing.[46] CNNs are more effective in classifying Raman spectra and show improved sensitivity and specificity for mixture analysis compared to other ML methods, including k-NN, SVM, and RF.[47] Compared to fully connected neural networks, CNN can decrease the number of filters and thus reduce its complexity. The convolutional nature of CNN also makes it insensitive to translations in the spectra from instrumental and environmental differences.

A potential problem with deep learning, including CNNs, is their need for large sets of training data, which can be impractical or difficult to obtain. However, when the SERS spectra of each mixture component are known, training data can be simulated by virtually mixing SERS spectra of individual components. Zhu et al. used such simulated datasets to train a CNN for pesticide detection utilizing SERS.[47] To the best of our knowledge, CNNs using simulated datasets have not previously been explored for spectral unmixing analysis of multiplexed, dye-labeled SERS spectra. We demonstrate for the first time that CNN trained with simulated datasets can successfully decompose multiplexed SERS spectra from mixtures of seven distinct dye-labeled, SERS-active metallic nanoparticles. This study compares CNN performance to that of SD, PLSR, random forest regression (RFR), and support vector regression (SVR). We then demonstrate general CNN, PLSR, SVR, and RFR methods that can be applied to decompose any multiplexed spectrum composed of known component spectra. We apply the method to SERS particle assays using ultrabright nanorattles to detect a single mRNA biomarker for head and neck cancer in 20 patients' samples. The results demonstrate the use of multiple ML models for spectral unmixing of SERS spectra. Although clinical data used here are not multiplexed, comparative analysis of singleplexed data using a CNN, SVR, RFR, and PLSR trained on seven-way multiplexed simulated spectra demonstrates robustness of CNN against false positives and shows a proof-of-concept in using multiple models for SERS-based medical diagnostics.

2 | METHODS

2.1 | Synthesis and clinical procedures

2.1.1 | Nanorattle synthesis

Gold-coated silver nanorattles were synthesized as reported previously.[16] Briefly, gold nano-particles (GNPs) were prepared using a seed-mediated method, resulting in 20 nm GNPs. A silver shell was coated on the GNPs by reducing AgNO3 with ascorbic acid in the presence of cetyltrimethylammonium chloride (CTAC), yielding gold–silver core-shell structures (GNP@ AgCubes). The silver shells were then converted into cages using galvanic replacement. For Raman reporter loading, the stock GNP@AgCages were mixed with seven distinct Raman dyes: indocyanine green (ICG) at 50 uM, 3,3′-diethylthiatricarbocyanine (DTTC) at 50 uM, 1,1′,3,3,3′,3′-hexamethylindotricarbocyanine (HITC) at 50 uM, IR775 at 1 mM, IR780 at 1 mM, IR792 at 1 mM, and IR7971 at 1 mM in ethanol. These dyes were chosen because their fluorescence excitation is around the laser excitation of 785 nm to produce SERRS for further signal enhancement. After 2 h under shaking, the suspension was washed for the final gold coating, which was performed by reducing gold chloride with ascorbic acid in the presence of CTAC. Nanorattle synthesis and assay procedure is summarized in Figure 1.

2.1.2 | Clinical samples

Tissue samples were obtained and assay was conducted as previously described.[20] Briefly, patients undergoing head and neck surgery at Duke University including head and neck squamous cell carcinoma, thyroid papillary carcinoma, lymphoma, benign lymphoid, or tonsillar disease were eligible for the study. Fresh tissue samples were collected intraoperatively, dissected for histopathology, flash

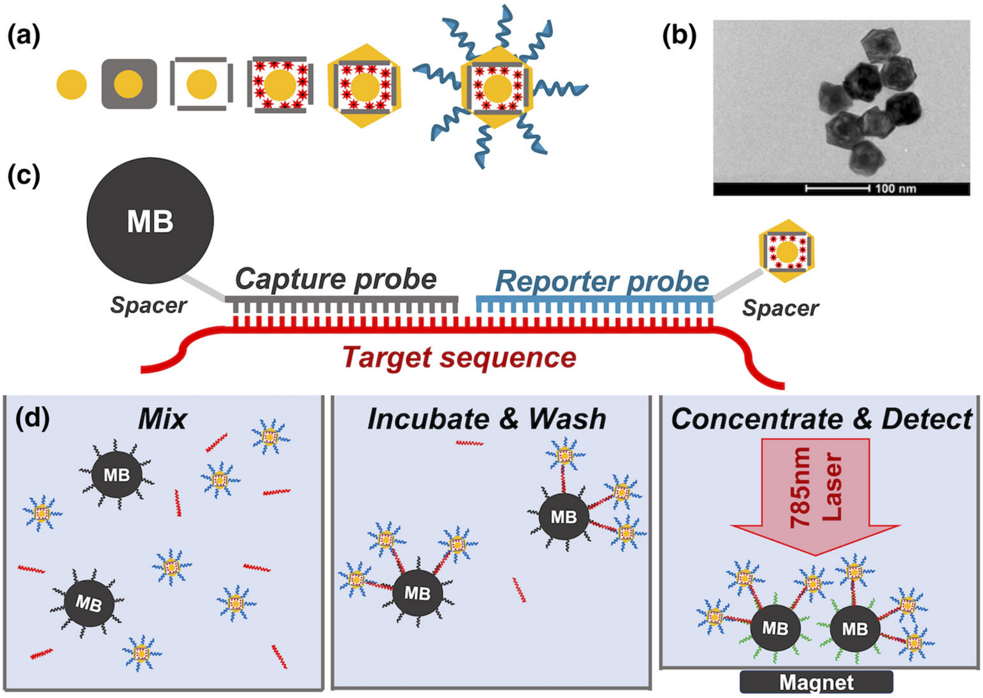

FIGURE 1 (a) Steps in nanorattle synthesis: 20 nm Au spheres, growth of Ag cube, galvanic replacement resulting in Au@Ag cage, dye loading, final Au coating, and DNA probe functionalization. (b) TEM of nanorattle. (c) Hybridization scheme of nanorattle and magnetic bead hybridization assay. (d) Nanorattle assay steps: mix magnetic beads, nanorattles, and target; incubate; concentrate; and detect. TEM, transmission electron microscopy [Colour figure can be viewed at wileyonlinelibrary.com] frozen in liquid nitrogen, and stored at −80°C. RNA was extracted from samples using Qiagen RNAeasy Mini Kit. Nanorattles were synthesized as described above, loaded with HITC (1 mM), gold coated, and functionalized with thiolated DNA probes for cytokeratin 14 (CK14) as previously described.[19] Magnetic beads (Dynabeads MyOne Streptavidin C1, 1-μm diameter) were functionalized with biotinylated DNA capture probe according to manufacturer's protocol. DNA detection was conducted using RNA extract, functionalized magnetic beads, and functionalized nanorattles in PBS + 0.01% Tween-20. Samples were incubated at 40°C for 3 h, washed 3X, and transferred into capillary tubes. Nanorattle-magnetic bead complexes were concentrated by a magnet. SERS was measured at 300 mW with 1 s exposure time using a lab-built 785 nm laser system including a Rigaku Xantus-1 laser, InPhotonics RamanProbe, Princeton Instruments Acton LS 785 spectrometer, and Princeton Instruments PIXIS: 100BR_eXcelon CCD camera.

2.2 | Data augmentation strategy

2.2.1 | Reference acquisition for data augmentation

Dye-loaded nanorattle solution of 5 ul was loaded into a glass capillary tube for each distinct dye. Spectra in the range 278–2383 cm^{-1} were taken using above laser system at 100 mW power, 100 ms exposure time, and with 10 accumulations (Figure 2).

2.2.2 | Data preprocessing

The 278–1627 cm^{-1} range contained rich SERS features and were used for training and testing. Each raw reference spectrum was smoothed and background-subtracted using a Savitzky–Golay filter, with five-point window and first-order polynomial, on MATLAB.

2.2.3 | Training set and label generation

Simulated spectral mixtures were made by mathematically combining reference spectra. Four reference spectra were acquired and used for each distinct dye-loaded nanorattle. During data augmentation, one spectrum is randomly selected for use in mixture simulation. Data augmentation consisted of seven steps: (1) separate the background from raw spectrum; (2) normalize raw spectrum by dividing by the max intensity of the background subtracted spectrum; (3) scale normalized raw spectra by the simulated scaling factor (label); (4) horizontally shift spectra a random number of spaces left or right in the range [0 4]; (5) add Gaussian noise at random power in the range [3e-5 5e-3]; (6) subtract background in custom code with window 40; and (7) normalize background subtracted data by dividing by height of highest peak, so all spectral points are in the range [0 1]. Example of simulation steps is shown in Figure S6 (supporting information). Label for each training spectrum consisted of a vector of scaling factors for each normalized reference spectrum. Labels were generated by creating a simulated calibration curve of one spectrum, with concentrations of all other references generated randomly such that sum of all labels equaled 1. This was repeated until a calibration curve was simulated for each distinct reference. All data were augmented in MATLAB 2018b.

2.2.4 | Validation set generation

A validation set of 2000 spectra were simulated using the same method as used above for training dataset.

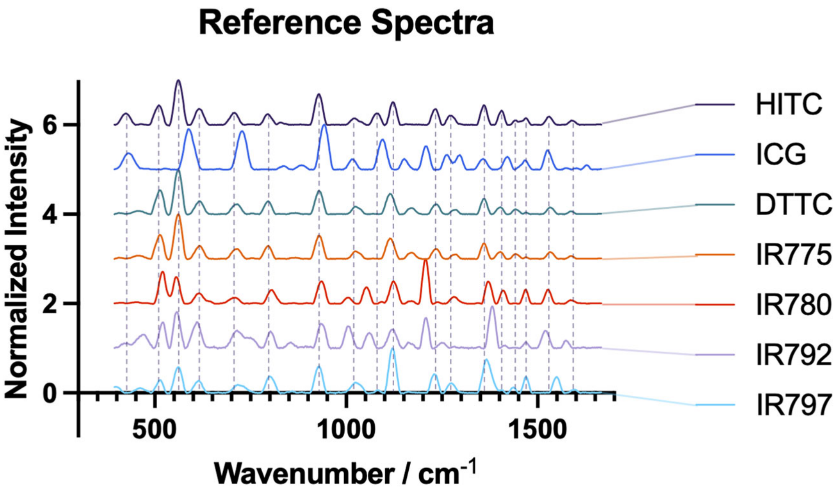

FIGURE 2 Averaged, background subtracted, and normalized reference spectra of nanorattles loaded with seven distinct Raman-active dyes. Spectra are offset for visual clarity. Vertical dashed lines show location of SERS peaks in HITC spectrum. SERS, surface-enhanced Raman spectroscopy; HITC, 1,1′,3,3,3′,3′-hexamethylindotricarbocyanine [Colour figure can be viewed at wileyonlinelibrary.com]

2.3 | Comparative spectral analysis of models using fivefold cross validation

2.3.1 | Partial least squares regression

PLSR is a multivariate, statistical method widely used in different types of spectroscopies. PLSR projects the data onto new bases, calculates a linear regression between X and Y, and then finds the direction in the new X space that explains the most variance in Y space. PLSR is especially useful when X values are not independent and have multicollinearity, as is the case with Raman spectra. PLSR analysis was conducted in Python scikit-learn package using 29 components and tolerance of 5.14×10^{-6}.

2.3.2 | Support vector regression

SVR is a regression technique related to SVM, a classification technique. In SVM, a kernel is used to map the data onto a higher dimensional "feature" space to find a hyperplane or separating line between two distinct classes in feature space. SVR is the regression equivalent of SVM, where the hyperplane is now a line of best fit between the predictor and output. Unlike linear regression, the best fit is not calculated by squared error; decision boundaries on either side of the hyperplane allow an error tolerance. SVR analysis was done in Python using scikit-learn with a radial basis function kernel, epsilon $= 1.03 \times 10^{-3}$, $C = 1.07$, gamma $= 2.54 \times 10^{-2}$.

2.3.3 | Random forest regression

RFR is an ensemble learning regression technique using decision trees and modified boostrap aggregation (bagging). Ensemble learning combines the output of multiple weak learners to produce a strong learner. RFR uses a modified bagging method which randomly samples the dataset with replacement to produce multiple subsets of the data. Each tree is grown using a different random sampled dataset. During the growth of each tree, only a random subset of predictors are considered for each split, increasing tree variability. These multiple decision trees produce outputs that are then averaged to achieve a more accurate final result. Assuming the error of each tree is independent, their averaged output would be more accurate than each individual output, with reduced variance and protection against overfitting. RFR analysis was done in Python using scikit-learn with max depth of 18, 36 max features, min sample leaf of 1, min sample split of 4, and 483 estimators.

2.3.4 | Convolutional neural network

The 1D-CNN was built using TensorFlow in Python and is comprised of an input layer, four 1D convolutional layers, separated by three max pooling layers, followed by a fully connected dense layer and output layer (Figure 3). The preprocessed SERS spectra is fed to the input layer, which passes to the first convolutional layer comprised of 16 kernels of size 12. A convolutional layer moves a kernel over the spectrum with a stride of 1 and outputs a feature map that is fed to a ReLU nonlinear activation function. A max pooling layer with stride 2 reduces dimensionality of the previous layer, decreasing risk of overfitting, and computational burden. After four convolutional and four max pooling layers, the data are flattened before being fed into the dense layer with a dropout rate of 10%. Dropout is a regularization technique that randomly omits certain nodes during training (at a set probability), reducing chances of overfitting. Finally, the dense layer goes to the output layer, which outputs the network's predictions, consisting of seven labels, each corresponding to the relative contribution of one reference.

The Adam optimizer in TensorFlow was used to compile the CNN with loss as mean squared error between predicted and true labels. All input arguments to Adam optimizer were kept at default values without further optimization.

2.3.5 | PLSR, SVR, and RFR hyperparameter optimization

PLSR, SVR, and RFR were optimized using Optuna for $n = 30$ trials. An optimization dataset of $n = 5600$ was generated using above method. Hyperparameter tuning ranges and distributions for all algorithms are available in supporting information (Table S1–S3 and Figures S1, S3, and S4).

2.3.6 | CNN architecture optimization

CNN architectures with varying numbers of convolutional layers and dropout rates were tested by comparing five-fold cross validation loss (Tables S5 and S6) in supporting information. CNN with four convolutional layers yielded the best performance and was utilized for further investigation. Four hyperparameters were additionally tuned in Optuna for further optimization (Table S4 and Figure S5 in supporting information).

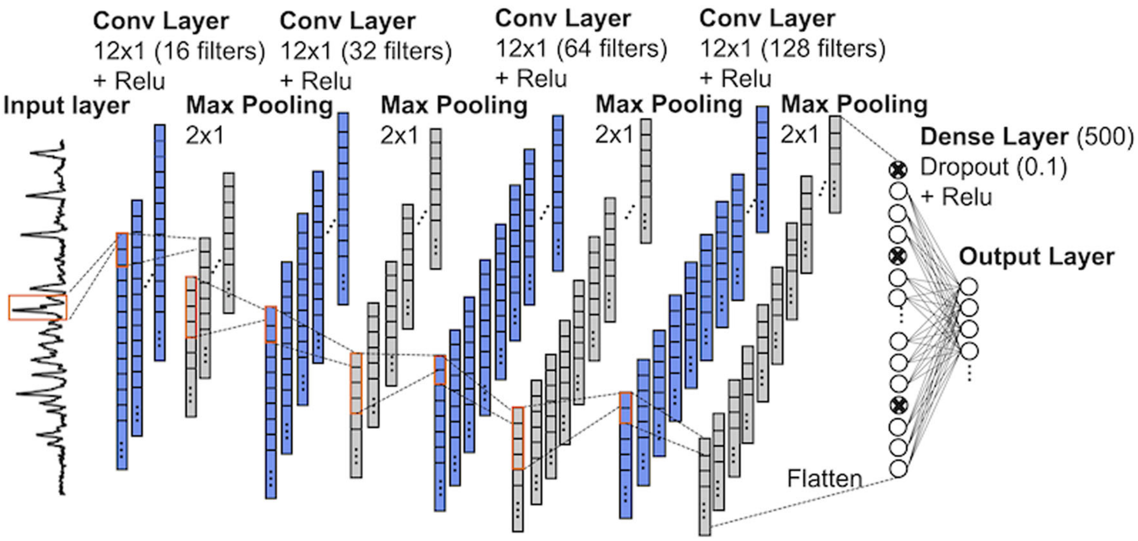

FIGURE 3 CNN layer architecture. CNN, convolutional neural networks [Colour figure can be viewed at wileyonlinelibrary.com]

2.3.7 | Fivefold cross validation (CV)

PLSR, SVR, RFR, and CNN models trained on $n = 16000$-simulated spectral training set were validated using shuffled, fivefold CV in scikit-learn. In fivefold CV, the data are randomly split into five, equal groups. One of which is used for validation, whereas the other four are used in training. This is repeated for each group, resulting in five total models. Root-mean-squared error, fit time, and prediction time per prediction were reported for each model. The mean and standard deviation of each were reported in Table 1. A one-way analysis of variance (ANOVA) was conducted on $\mathrm{RMSE_{label}}$, and Tukey's post hoc testing was used to determine statistical significance among algorithm performance, training time, and prediction time (Table S8 in supporting information).

$$\mathrm{RMSE_{label}} = \sqrt[2]{\frac{1}{n*m}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\hat{y}\mathrm{label}_{i,j} - Y\mathrm{label}_{i,j}\right)^2} \quad (1)$$

Equation (1) defines $\mathrm{RMSE_{label}}$, the performance metric used for each algorithm in fivefold cross validation. $\hat{y}\mathrm{label}_{i,j}$ and $Y\mathrm{label}_{i,j}$ denote the predicted and ground truth labels, respectively where labels corresponding to relative contribution of each reference in one simulated spectrum; i denotes indexing of sample number from 1 to n; and j denotes indexing of labels for each sample from 1 to $m = 7$. This metric was used to quantify error when ground truth labels were *known*.

2.4 | Comparative spectral analysis using multiplexed SERS spectral measurements

2.4.1 | SERS measurement of multiplexed nanorattles

Ten random mixtures were created by combining a random volume in the range 1–5 uL of each reference nanorattle solution. The resultant mixture was pipetted into a glass capillary tube and measured using the same 785 nm laser source stated above. Each mixture was measured at 80 and 100 mW to generate two test sets: one with high noise and one with low noise. Data were preprocessed as described above.

2.4.2 | Spectral decomposition

SD assumes that the raw mixture spectrum is made up of different contributions from reference spectra and an unknown polynomial (background). Therefore, SD was the only method that did not require spectral preprocessing by background subtraction, which is an advantage over all other methods discussed in this work. Each reference spectrum was measured by averaging of four measurements, normalized, and the entire collected spectral range 278–2383 $\mathrm{cm^{-1}}$ was used for analysis. Decomposition calculations were conducted in MATLAB using the lsqr and fmincon functions to find the best fit of the seven reference spectra and a free-fitting polynomial to the normalized, raw mixture spectrum. MATLAB algorithm generates the minimally constrained coefficients (i.e. fractional contribution) for each reference spectrum and the polynomial background. Finally, it outputs a normalized signal fraction for each reference and a polynomial.

TABLE 1 $RMSE_{label}$ (equation 1) and fit times for fivefold cross validation for PLSR, SVR, and CNN. $RMSE_{spectrum}$ (equation 2) and prediction times for SD, PLSR, SVR, and CNN predictions of SERS spectra obtained from seven-way multiplexed, dye-loaded nanorattles.

	SD	PLSR	SVR	RFR	CNN
Fivefold CV $RMSE_{label}$	n/a	$4.41 \times 10^{-2} \pm 3.55 \times 10^{-4}$	$2.16 \times 10^{-2} \pm 1.49 \times 10^{-4}$	$4.12 \times 10^{-2} \pm 2.81 \times 10^{-4}$	$2.12 \times 10^{-2} \pm 3.08 \times 10^{-4}$
Training time/s	n/a	$4.56 \pm 2.89 \times 10^{-1}$	$1.03 \times 10^{3} \pm 1.15 \times 10^{1}$	$1.70 \times 10^{2} \pm 6.86 \times 10^{-1}$	$1.81 \times 10^{3} \pm 8.43$
80-mW test set $RMSE_{spectrum}$	1.05×10^{-1}	8.51×10^{-2}	8.38×10^{-2}	8.89×10^{-2}	8.24×10^{-2}
100-mW test set $RMSE_{spectrum}$	4.12×10^{-2}	3.05×10^{-2}	2.32×10^{-2}	3.42×10^{-2}	2.34×10^{-2}
Prediction time/s	$7.77 \times 10^{-1} \pm 1.01 \times 10^{-1}$	$9.97 \times 10^{-5} \pm 3.88 \times 10^{-7}$	$6.05 \times 10^{-2} \pm 8.14 \times 10^{-4}$	$4.84 \times 10^{-4} \pm 2.11 \times 10^{-5}$	$7.63 \times 10^{-3} \pm 3.03 \times 10^{-3}$

Note: Test sets consisted of 2 sets of 10 spectra obtained using 80- or 100-mW laser power.
Abbreviations: CNN, convolutional neural networks; SVR, support vector regression; PLSR, partial least squares regression; RFR, random forest regression; SERS, surface-enhanced Raman spectroscopy; SD, spectral decomposition; CV, cross validation.

2.4.3 | CNN, PLSR, SVR, and RFR

CNN model was trained on $n = 16000$ simulated spectral training set with batch size of 2000 and validation set of $n = 3000$. PLSR, SVR, and RFR models were fit using $n = 16000$ simulated spectral training set without changing model parameters from previous fivefold cross validation and used for label prediction.

2.4.4 | Prediction evaluation for all algorithms

For evaluation of nanorattle mixture test sets, model predicted labels for each reference dye were used to simulate a predicted spectrum using reference spectra in a procedure similar to data augmentation: (1) separate the background and background subtracted portions of raw spectrum, (2) normalize both portions by dividing by the maximum intensity of the background subtracted spectrum, and (3) scale both normalized portions by the simulated scaling factor (label) and sum. Model performance was evaluated by root-mean-squared error of prediction ($RSME_{spectrum}$) between the actual and predicted spectral values in the spectra preprocessed by smoothing and fluorescence background subtraction. This was distinct from the fivefold validation $RMSE_{label}$, which is calculated from labels. Predictions were evaluated on both the 80-mW test set (lower signal to noise ratio) and the 100-mW test set (higher signal-to-noise ratio). Prediction times for both test sets were averaged and divided by number spectra per test set to acquire the average prediction time per spectrum.

$$RMSE_{spectrum} = \sqrt[2]{\frac{1}{n*w}\sum_{i=1}^{n}\sum_{k=1}^{w}\left(\hat{y}spectrum_{i,k} - Yspectrum_{i,k}\right)^2} \quad (2)$$

Equation 2 defines $RMSE_{spectrum}$, the performance metric used for each algorithm in performance evaluation of the test sets. $\hat{y}spectrum_{i,j}$ and $Yspectrum_{i,j}$ denote the predicted and ground truth spectral values, respectively, where spectral values correspond to each value in the measured SERRS spectrum; i denotes indexing of sample number from 1 to n; and k denotes indexing of number of spectral values for each sample from 1 to $w = 700$. This metric was used to quantify error when ground truth labels were *unknown*.

2.4.5 | Varied training set size and smoothing

Final training datasets excluding validation set consisted of 70, 350, 700, 1400, 2800, 5600, and 16000 spectra. The simulated raw spectra were preprocessed by background subtraction with and without smoothing. The resulting training sets were used for training of CNN, SVR, RFR, and PLSR models with varying set size and with or without smoothing. Validation sets were used for each varied dataset corresponding to a 70/30 training/validation split to prevent overfitting. Model was trained using batch size equal to training set size for smaller training sets less than 4000, and batch size was 2000 for training sets larger than 4000 training sets. Batch size was determined by trial and error. In the case of RFR and CNN, where models have inherent randomness, five models were trained for each condition to obtain a mean and standard deviation of $\text{RMSE}_{\text{spectrum}}$ (Equation 2) using smoothed references. For SVR and PLSR, one model was trained for each condition. Training time and time per prediction were recorded for each model.

2.5 | Application to clinical samples

2.5.1 | Clinical data processing

The clinical dataset consisted of duplicate spectra for each of the $n = 20$ samples. Each spectrum was obtained from a distinct assay using singleplex HITC-loaded nanorattles to detect CK14 mRNA. Each raw spectrum was preprocessed by smoothing and background subtraction as described above. The spectrum was segmented to include only 770–1627 cm^{-1}. This region was selected to exclude regions with complex background (280–770 cm^{-1}) and regions without SERS peaks (1630–2400 cm^{-1}). All spectra were normalized to the maximum spectral value. Duplicates for each sample were averaged before further analysis.

2.5.2 | Clinical data analysis

CNN, SVR, RFR, and PLSR were trained using simulated spectra $n = 16000$ mixtures including all seven references. HITC reference contribution to simulated mixtures consisted of two blank HITC spectra and two positive control HITC spectra from clinical assay. An example 10-nM synthetic DNA target was used to obtain the positive control assay spectra. The negative control spectra were obtained from extracted RNA from a negative sample. The correct output label is known because only one dye is present (one for HITC label and zero for all other dyes). Predicted labels were normalized to $\boldsymbol{\Sigma}\bar{y} = 1$, and accuracy was then determined by $\text{RMSE}_{\text{label}}$ between normalized predicted labels and actual labels (Equation 1).

3 | RESULTS AND DISCUSSION

3.1 | Fivefold cross validation

Cross validation was performed for PLSR, SVR, RFR, and CNN on $n = 16000$ training data set. CNN performed the best in fivefold cross validation with the lowest $\text{RMSE}_{\text{label}}$ (Table 1). PLSR and RFR performed much worse than CNN and SVR. CNN and SVR performed with similar accuracy. One-factor ANOVA on $\text{RMSE}_{\text{label}}$ between the three models resulted in a significant p value $<2 \times 10^{-16}$, which is much smaller than the alpha value $p = 0.05$. The significant ANOVA allowed post hoc Tukey's testing, which revealed that PLSR and RFR performed significantly worse than CNN and SVR, RFR performed significantly better than PLSR, and there was no significant difference in the performance of CNN versus SVR ($p = .143$). See Table S8 in supporting information for more detailed statistical test results.

It is noteworthy that CNN did not perform significantly better than SVR. An adaptive learning rate could be employed over the default constant learning rate in CNN to potentially improve performance.

All models had significantly different training times. CNN required the longest training time of ∼30 min, followed by SVR with a training time of ∼17 min. RFR required less than 3 min for training, and PLSR had the fastest training time of only ∼5 s (Table 1).

3.2 | Validation on multiplexed test sets of varying S/N

For all algorithms, $\text{RMSE}_{\text{spectrum}}$ was lower for 100-mW test set because of the higher signal-to-noise ratio compared to the 80-mW test set (Table 1). Since the predicted spectra were reconstructed using smoothed and averaged references, noise will contribute to difference between the predicted and test spectra, increasing error. Therefore, $\text{RMSE}_{\text{spectrum}}$ cannot be compared between the two datasets but can be compared within each test set. The projected, predicted spectrum fits very well with input test spectrum, and the difference between the predicted and test spectra showed that discrepancies in the prediction were largely within the magnitude of noise (Figure 4). See Figures S12–S15 in supporting information for more plots of predicted versus test spectra.

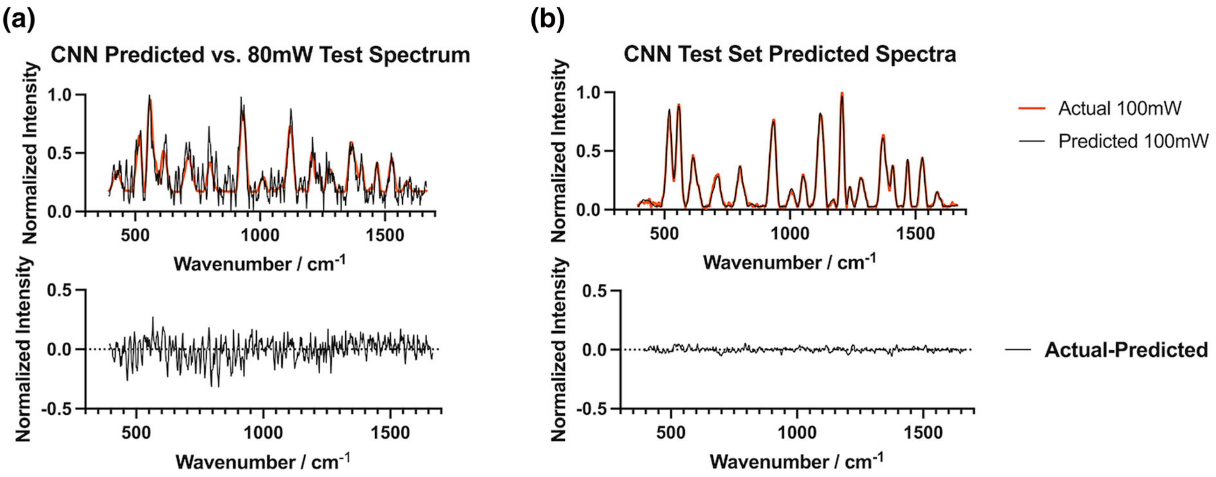

FIGURE 4 Example comparisons of predicted spectra reconstructed from CNN and test spectra from different datasets. (a) Top: Test spectrum from 80-mW test set is plotted in black and predicted spectrum from CNN is plotted in orange. Bottom: Difference between the predicted and actual spectrum is plotted to show error. (b) Top: Test spectrum from 100-mW test set is plotted in black, and predicted spectrum from CNN is plotted in orange. Bottom: Difference between the predicted and actual spectrum is plotted to show error. CNN, convolutional neural networks [Colour figure can be viewed at wileyonlinelibrary.com]

CNN had the lowest $\text{RMSE}_{\text{spectrum}}$ for the 80-mW dataset. SVR and CNN had similar $\text{RMSE}_{\text{spectrum}}$ for the 100-mW dataset (Table 2), which were lower than other model types. SD performed the worst for both datasets (Table 1). For the 80-mW dataset, $\text{RMSE}_{\text{spectrum}}$ of SD was 1.61×10^{-2} greater than the third best performing model, RFR, whereas for the 100-mW dataset, $\text{RMSE}_{\text{spectrum}}$ of SD was only 7.00×10^{-3} greater than RFR. The increased error of SD predictions in 80-mW test set compared to performance of other algorithms shows SD did not perform well with low S/N spectra. However, in low S/N spectra, SD performed well compared to other algorithms and offers benefits of not requiring training and preprocessing (background subtraction), which can introduce their own errors.

In both datasets, SVR performed better than RFR and PLSR. SVR produced the highest accuracy predictions in the 100-mW test set, whereas CNN produced the highest accuracy predictions in the 80-mW test set. These results match well with results from the fivefold cross validation. SVR and CNN performed similarly during cross-fold validation, and SVR slightly outperformed CNN in 100-mW test set but performed worse in the 80-mW test set (Table 1). There will inevitably be slight differences between simulated training data and acquired experimental data, so the robustness and generalizability of the model are important considerations. CNN was more robust to noise compared to SVR, which may be because of the convolutional nature of the model. By learning general shapes and relative spectral values instead of absolute values, CNN is more robust to small changes and shifts in spectra between simulated and experimentally acquired spectra.

Prediction speed for the two datasets were compared in Table 1. SD had a significantly slower spectrum prediction speed compared to all other models. There was no statistically significant difference between prediction times in SVR, PLSR, RFR, or CNN (Table S8 in supporting information), potentially because of small size of test sets. SVR had the second longest prediction time followed by CNN, RFR, and then PLSR.

3.3 | Effect of varying training set size on performance

Next, the training set size of smoothed training sets was varied to investigate the relationship between training set size and algorithm performance (Figure 5a, solid markers). SVR and CNN performed best in the 100-mW test set, with the lowest $\text{RMSE}_{\text{spectrum}}$ (Figure 5a). RFR performed the worst with the highest $\text{RMSE}_{\text{spectrum}}$. In smaller training sets, SVR performed better than CNN, until around $n = 2000$ training set, where CNN performed comparably to SVR. For performance on the 80-mW test set, CNN always outperformed SVR, demonstrating that CNN is more robust to noise than SVR. Again, RFR had the highest $\text{RMSE}_{\text{spectrum}}$. As training set size decreased, CNN, SVR, and RFR prediction accuracy generally became worse with more variability. CNN and SVR performance plateaued at a certain training set size ($n = 4000$ for CNN and $n = 8000$ for SVR), whereas RFR

TABLE 2 $\mathrm{RMSE_{label}}$ performance of PLSR, SVR, RFR, and CNN predictions on clinical singleplex data (only HITC-loaded particles present)

	$\mathrm{RMSE_{label}}$			
Dye label	**PLSR**	**SVR**	**RFR**	**CNN**
HITC	3.24×10^{-1}	3.24×10^{-1}	3.24×10^{-1}	1.48×10^{-1}
DTTC	4.70×10^{-2}	4.70×10^{-2}	4.70×10^{-2}	1.51×10^{-2}
ICG	4.35×10^{-2}	4.35×10^{-2}	4.70×10^{-2}	1.61×10^{-2}
IR775	9.80×10^{-2}	9.80×10^{-2}	9.80×10^{-2}	6.04×10^{-2}
IR780	3.80×10^{-2}	3.80×10^{-2}	3.80×10^{-2}	2.27×10^{-2}
IR792	1.91×10^{-2}	1.91×10^{-2}	1.91×10^{-2}	0.00
IR797	1.13×10^{-1}	1.13×10^{-1}	1.13×10^{-1}	4.88×10^{-2}
All labels	6.54×10^{-2}	1.38×10^{-1}	1.07×10^{-1}	6.42×10^{-2}

Note: $\mathrm{RMSE_{label}}$ for each dye is shown separately and total $\mathrm{RMSE_{label}}$ is shown.

Abbreviations: CNN, convolutional neural networks; SVR, support vector regression; PLSR, partial least squares regression; RFR, random forest regression; HITC, 1,1′,3,3,3′,3′-hexamethylindotricarbocyanine; DTTC, 3,3′-diethylthiatricarbocyanine; ICG, indocyanine green.

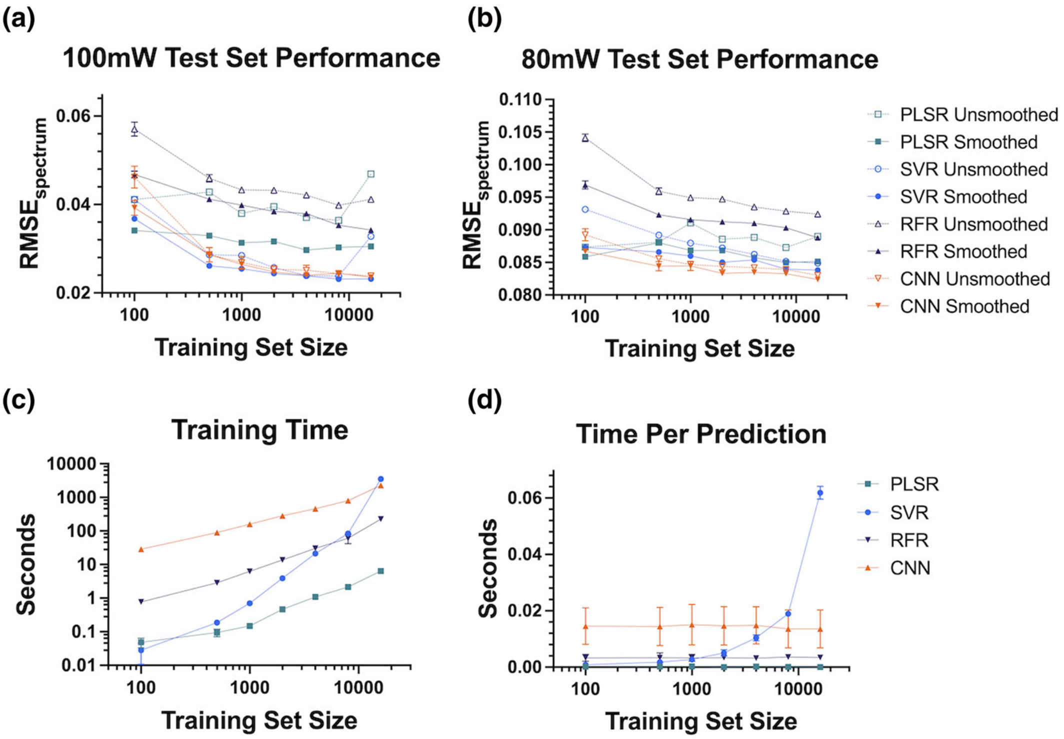

FIGURE 5 RMSEspectrum performance of $n = 5$ CNN, $n = 5$ RFRs, SVR, and PLSR on 100-mW test set (a) and 80-mW test set (b) are plotted as circular markers for training sets of size $n = 100$, $n = 500$, $n = 1000$, $n = 2000$, $n = 4000$, $n = 8000$, and $n = 16000$ (70% training, 30% validation) on a semilog plot. Smoothed training sets are plotted with solid markers, connected by solid lines, and unsmoothed training sets are plotted as hollow markers, connected by dashed lines. Training time (c) and time per prediction (d) for PLSR, SVR, RF, and CNN were averaged between models trained on smoothed and unsmoothed data with above-mentioned training set sizes and plotted on a log-log and semi-log plot, respectively. CNN, convolutional neural networks; SVR, support vector regression; PLSR, partial least squares regression; RFR, random forest regression [Colour figure can be viewed at wileyonlinelibrary.com]

performance on 100-mW test set continued to improve with larger training sets. Because of poor performance accuracy and need for large training datasets, RFR may not be suitable for this task. Meanwhile, PLSR performance on both test sets trended very weakly with training set size. Because of its structure, PLSR performe well with small training sets, and increasing training set size using the same reference spectra did not provide additional information that would improve performance.

The effect of smoothing on model performances was investigated at different training sizes (Figure 5a, b hollow markers). Models trained with unsmoothed training sets performed worse or similar to their counterparts trained with smooth data. For CNN, SVR, and RFR models, difference in performance between models trained with smooth and unsmooth training sets decreased as training set size increased, showing the importance of data smoothing especially for smaller training sets. For CNN models tested on the 100-mW test set, as the training set size increased, the discrepancy in $\mathrm{RMSE_{spectrum}}$ between models trained on smoothed and unsmoothed data diminished quickly (Figure 5a). CNN quickly learned the noise in high signal-to-noise spectra. However, in the 80-mW dataset, as the training set size increased, the discrepancy in performance diminished but was still present (Figure 5b). A possible cause for this discrepancy was errors introduced by background subtraction of unsmoothed data. As smoothing occured first, it affected the background subtraction process and thus very noisy data resulted in less effective fluorescence background removal. Although CNN can learn noise, smoothing improved background subtraction and decreased size of training set needed to achieve better performance.

There were two outliers in the performance of PLSR and SVR models trained on unsmoothed spectra and tested on the 100-mW test set. At $n = 16000$, PLSR and SVR performed with much higher $\mathrm{RMSE_{spectrum}}$ than their counterpart models trained on smaller datasets, suggesting that SVR and PLSR were less robust to noise.

Training time versus training set size was compared to investigate the tradeoff between model accuracy and training time for increasing training set sizes (Figure 5c). For all models, smaller training sets required less computing power and time during training. Although training times for CNN, SVR, and PLSR increased at similar rates with increasing training set size, SVR training times increased much faster. Thus, SVR was not suitable for large datasets above a $n = 10000$. CNN models provided diminishing returns above a training set size of $n = 4000$ as the small improvement in accuracy did not justify the larger cost of memory and computational power. Finally, time per prediction was compared in models with different training set sizes. For CNN, PLSR, and RFR, prediction time did not depend on training set size (Figure 5d). PLSR was consistently the fastest predictor followed by RFR then CNN. However, for SVR, prediction time increased exponentially with increasing training set size, starting at similar speeds to PLSR at the smallest set size ($n = 100$) to becoming the slowest predict at larger set sizes ($n \geq 8000$).

For CNN, there were diminishing returns on accuracy above a certain training set size. Based on characterization of $\mathrm{RMSE_{spectrum}}$ versus training set size for this CNN architecture and spectral unmixing problem using smoothed training data (Figure 5a, orange markers), 2000 total spectra consisting of 70% training and 30% validation achieved accurate results without excessive cost in computing power, time, and memory. Although SVR performed well in the 100-mW test set trained at the high training set size, its rapid increase in training and prediction time with increasing training set size made it less suitable for use with larger training sets. Compared to CNN, SVR was also less robust to noise. RFR performed with the worst accuracy and was not suitable for this application. PLSR performed the best out of all the models in the smallest size training set ($n = 100$) and performed the be for small training sets.

3.4 | Analysis of clinical data

Twenty clinical samples were previously analyzed[20] for CK14 mRNA using the singleplexed nanorattle assay with HITC-loaded nanorattles. The model trained on seven reference dyes was used for spectral unmixing of a clinical dataset composed of preprocessed SERS spectra from clinical assays. Because the clinical assay included only HITC-loaded nanorattles, the ground truth labels of spectral contribution were known to be 1 for HITC and 0 for all other dyes. Thus, loss was defined as $\mathrm{RMSE_{label}}$. $\mathrm{RMSE_{label}}$ results from PLSR, SVR, RFR, and CNN were compared in Table 2. CNN performed best with the lowest $\mathrm{RMSE_{label}}$. Interestingly, PLSR performed with similar accuracy to CNN, whereas SVR and RFR had comparatively much worse $\mathrm{RMSE_{label}}$.

For all models, $\mathrm{RMSE_{label}}$ was lowest for dye labels corresponding to nanorattles loaded with IR792. No incorrect CNN predictions were made for nanorattles loaded with IR792 (Table 2). The lower rates of incorrect predictions for ICG- and IR792-loaded nanorattles are likely because, within the analyzed spectral wavenumbers (770–1627 $\mathrm{cm^{-1}}$), their SERS spectra have distinct peaks that do not overlap with SERS peaks obtained from HITC-loaded nanorattles. ICG- and IR792-loaded nanorattles exhibited SERS reference spectra with three peaks that do not overlap with peaks from HITC-loaded nanorattle reference spectra. SERS reference spectra of ICG-loaded nanorattles have distinct peaks located at 944.8, 1096, and 1155 $\mathrm{cm^{-1}}$ (Figure 2). SERS reference spectra of IR792-loaded nanorattles have distinct peaks located at 1005, 1209, and 1383 $\mathrm{cm^{-1}}$. On the other hand, SERS reference spectra from DTTC-, IR775-, IR780-, and IR797-loaded nanorattles had peaks that all completely or partially overlapped with HITC-loaded SERS reference spectra within the analyzed spectral range (Figure 2). The similarity in spectral shape increased the difficulty of spectral unmixing. The dependence between shape similarity of SERS reference spectra and prediction performance of the model highlights the importance of selecting dyes with maximum difference in SERS spectra shape.

3.5 | Algorithm tradeoffs and considerations

PLSR, SVR, RFR, and CNN were used for clinical data analysis. Table S9 in supporting information provides a general comparison of various performance criteria of the different ML techniques. Computing power, memory, and total power are an especially important consideration for POC applications. Ideal POC devices should be portable, easy to use, low cost, and power efficient. Integrated data analysis is necessary for applications with difficult-to-interpret data such as SERS. Portable microprocessors used for POC such as Raspberry Pi is limited in computing power and memory. Although SD, PLSR, and SVR generally predicted less accurately than CNN in multiplexed test sets, they required less computing power, memory, and training time. But once the CNN was trained, predictions we made very rapidly. Because models are trained offline while predictions are made online or at the point of care, prediction speed is a much more important consideration than training speed.

3.5.1 | Hyperparameter tuning

Prediction performance of CNN, SVR, and RFR models depend greatly on hyperparameter tuning. SVR greatly relies on tuning of gamma and epsilon to achieve optimal model performance. Important hyperparameters to be tuned in RFR include number of decision trees in forest, maximum depth of trees, minimum samples to split on at an internal node, etc. CNN has many more tunable hyperparameters than the other employed algorithms because of their many possible architectures (see Tables S4–S6 and Figures S5 in supporting information for details).

3.5.2 | Model interpretability

CNN is much less interpretable compared to SD and PLSR. SD is the most interpretable, as it is a straightforward fit of given references and a polynomial to the input spectrum. PLSR and SVR are less interpretable compared to SD, but much more interpretable than CNN. Important features of PLSR can be visualized by calculating the variable importance in projection (VIP) scores for the PLSR model, which estimate the importance of each variable, showing the wavelengths that contribute most to model predictions (Figure S2B in supporting information). Decision trees in RFR can be visualized for physical interpretation of the model (Figure S7 in supporting information), although deep trees are difficult to interpret. SVR visualization techniques are less straightforward, but the feature space and kernel can be visualized to correlate specific features responsible for certain output predictions.[48] Although CNN is touted as a "black box," visualization of filters and outputs after each filter can be used to make sense of CNN operation (see supporting information Figures S8–S11). In the case where reference spectral shapes were known, interpretability is less crucial. However, classification tasks based on non-dye-labeled spectra where spectral contributions are unknown may require more interpretable models such as the above-mentioned methods.

3.5.3 | Data preprocessing

SD does not require data simulation or training on large datasets; it only requires reference spectra for analysis, saving computation power and time. CNN, SVR, PLSR, and RFR are greatly affected by the preprocessing method because of their highly sensitive nature. Raw data are much more difficult to simulate correctly because of background variations arising from slight changes in assay conditions such as ambient light and laser focus. Training with background-subtracted data makes the model more generalizable to experiments conducted on different days and under varying conditions. Shifts and noise were added during data simulation procedure to account for spectral variations. It is important to note that background subtraction methods can be prone to errors, which can be minimized by preprocessing methods including smoothing. As seen above, smoothing improved model performance at smaller training set sizes, which decreased training time and required memory/computing power.

Despite CNN's disadvantages in memory/power consumption and low interpretability, its advantages in robustness to noise and variations, increased sensitivity, and fast predictions make it a powerful tool in multiplexed spectra analysis.

4 | CONCLUSION

In this study, we evaluated the performance of CNN and other common data analysis methods including PLSR, SVR, RFR, and SD on spectral unmixing of multiplexed SERS spectra using Raman-active dye labeled nanorattle probes. CNN analysis yielded the highest accuracy and fast predictions but required more computing power and memory to train the network using sizable datasets. CNN, SVR, PLSR, and RFR models were used to analyze clinical data of non-multiplexed mRNA biomarker analysis for head and neck cancer diagnostics using 20 patient samples used here as the model system for biomedical diagnostics. Compared to other models, CNN, trained on seven different SERS spectra, identified the specific clinical biomarker target for the disease with the lowest error, yielding $\text{RMSE}_{\text{label}} = 6.42 \times 10^{-2}$. The results demonstrate the capability of CNN-based ML for SERS-based medical diagnostics.

ACKNOWLEDGMENTS

This study is supported by the National Institutes of Health (R01GM135486, 1R01DE030455-01A1). Joy Li appreciates the support from NDSEG Fellowship (Fellow ID: 00007902).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Tuan Vo-Dinh https://orcid.org/0000-0003-3701-3326

REFERENCES

[1] P. A. Mosier-Boss, *Nanomaterials* **2017**, *7*, 142.

[2] M. Muhammad, Q. Huang, *Talanta* **2021**, *227*, 122188.

[3] T. J. Moore, A. S. Moody, T. D. Payne, G. M. Sarabia, A. R. Daniel, B. Sharma, *Biosensors* **2018**, *8*, 46.

[4] V. Cupil-Garcia, P. Strobbia, B. M. Crawford, H.-N. Wang, H. Ngo, Y. Liu, T. Vo-Dinh, *J. Raman Spectrosc.* **2021**, *52*, 541.

[5] T. Vo-Dinh, H.-N. Wang, J. Scaffidi, *J. Biophotonics* **2010**, *3*, 89.

[6] Y. Wang, J. Irudayaraj, *Philos. Trans. R. Soc. London, Ser. B* **2012**, *368*, 20120026.

[7] B. G. Zimmermann, D. T. Wong, *Oral Oncol.* **2008**, *44*, 425.

[8] L. Tribolet, E. Kerr, C. Cowled, A. G. D. Bean, C. R. Stewart, M. Dearnley, R. J. Farr, *Front. Microbiol.* **2020**, *11*, 1197.

[9] J. Viereck, T. Thum, *Circ. Res.* **2017**, *120*, 381.

[10] H. Marks, M. Schechinger, J. Garza, A. Locke, G. Coté, *Nanomedicine* **2017**, *6*, 681.

[11] H. N. Wang, A. M. Fales, A. K. Zaas, C. W. Woods, T. Burke, G. S. Ginsburg, T. Vo-Dinh, *Anal. Chim. Acta* **2013**, *786*, 153.

[12] T. Vo-Dinh, *TrAC Trends Anal. Chem.* **1998**, *17*, 557.

[13] H. N. Wang, B. M. Crawford, A. M. Fales, M. L. Bowie, V. L. Seewaldt, T. Vo-Dinh, *J. Phys. Chem. C. Nanomater. Interfaces* **2016**, *120*, 21047.

[14] K. Chen, M. Leona, K.-C. Vo-Dinh, F. Yan, M. B. Wabuyele, T. Vo-Dinh, *J. Raman Spectrosc.* **2006**, *37*, 520.

[15] H. N. Wang, A. M. Fales, T. Vo-Dinh, *Nanomedicine* **2015**, *11*, 811.

[16] H. T. Ngo, N. Gandra, A. M. Fales, S. M. Taylor, T. Vo-Dinh, *Biosens. Bioelectron.* **2016**, *81*, 8.

[17] N. Gandra, H. C. Hendargo, S. J. Norton, A. M. Fales, G. M. Palmer, T. Vo-Dinh, *Nanoscale* **2016**, *8*, 8486.

[18] H. T. Ngo, E. Freedman, R. A. Odion, P. Strobbia, A. S. De Silva Indrasekara, P. Vohra, S. M. Taylor, T. Vo-Dinh, *Sci. Rep.* **2018**, *8*, 4075.

[19] P. Dukes, H. Ngo, W. Lee, T. Dinh, *Anal. Methods* **2017**, *9*, 5550.

[20] P. V. Dukes, P. Strobbia, H. T. Ngo, R. A. Odion, D. Rocke, W. T. Lee, T. Vo-Dinh, *Anal. Chim. Acta* **2020**, *1139*, 111.

[21] J. Noonan, S. M. Asiala, G. Grassia, N. MacRitchie, K. Gracie, J. Carson, M. Moores, M. Girolami, A. C. Bradshaw, T. J. Guzik, G. R. Meehan, H. E. Scales, J. M. Brewer, I. B. McInnes, N. Sattar, K. Faulds, P. Garside, D. Graham, P. Maffia, *Theranostics* **2018**, *8*, 6195.

[22] B. R. Lutz, C. E. Dentinger, L. N. Nguyen, L. Sun, J. Zhang, A. N. Allen, S. Chan, B. S. Knudsen, *ACS Nano* **2008**, *2*, 2306.

[23] K. Gracie, E. Correa, S. Mabbott, J. A. Dougan, D. Graham, R. Goodacre, K. Faulds, *Chem. Sci.* **2014**, *5*, 1030.

[24] F. Nicolson, L. E. Jamieson, S. Mabbott, K. Plakas, N. C. Shand, M. R. Detty, D. Graham, K. Faulds, *Chem. Commun.* **2018**, *54*, 8530.

[25] R. Botta, P. Chindaudom, P. Eiamchai, M. Horprathum, S. Limwichean, C. Chananonnawathorn, V. Patthanasettakul, B. Kaewseekhao, K. Faksri, N. Nuntawong, *Tuberculosis* **2018**, *108*, 195.

[26] K.-M. Lee, T. J. Herrman, *Food Bioproc. Tech.* **2016**, *9*, 588.

[27] I. S. Patel, W. R. Premasiri, D. T. Moir, L. D. Ziegler, *J. Raman Spectrosc.* **2008**, *39*, 1660.

[28] H. Shin, H. Jeong, J. Park, S. Hong, Y. Choi, *ACS Sensors* **2018**, *3*, 2637.

[29] Y. Yu, Y. Lin, C. Xu, K. Lin, Q. Ye, X. Wang, S. Xie, R. Chen, J. Lin, *Biomed. Opt. Express* **2018**, *9*, 6053.

[30] B. Yan, B. Li, Z. Wen, X. Luo, L. Xue, L. Li, *BMC Cancer* **2015**, *15*, 650.

[31] S. Weng, B. Yuan, Z. Zhu, L. Huang, D. Zhang, L. Zheng, *Fast quantitative detection of thiram using surface-enhanced Raman scattering and support vector machine regression; SPIE*, Vol. 10255, SOE, Jinhua, Suzhou, Chengdu, Xi'an, Wuxi, China **2017**.

[32] S. Li, Y. Zhang, Q. Zeng, L. Li, Z. Guo, Z. Liu, H. Xiong, S. Liu, *Laser Phys. Lett.* **2014**, *11*, 065603.

[33] C. Jian, J. Boyan, Z. Ying, W. Zhenyu, *J. Appl. Spectrosc.* **2021**, *88*, 225.

[34] Y. Fan, K. Lai, B. A. Rasco, Y. Huang, *LWT- Food Sci. Technol.* **2015**, *60*, 352.

[35] R. Dong, S. Weng, L. Yang, J. Liu, *Anal. Chem.* **2015**, *87*, 2937.

[36] T. Rojalin, D. Antonio, A. Kulkarni, R. P. Carney, *Appl Spectrosc* **2021**, *76*, 37028211034543.

[37] W. Hu, S. Ye, Y. Zhang, T. Li, G. Zhang, Y. Luo, S. Mukamel, J. Jiang, *J. Phys. Chem. Lett.* **2019**, *10*, 6026.

[38] L. Shen, Y. Du, N. Wei, Q. Li, S. Li, T. Sun, S. Xu, H. Wang, X. Man, B. Han, *Spectrochim. Acta, Part a* **2020**, *237*, 118364.

[39] S. Seifert, *Sci. Rep.* **2020**, *10*, 5436.

[40] C. D. L. Albuquerque, R. B. Nogueira, R. J. Poppi, *Microchem. J.* **2016**, *128*, 95.

[41] N. Banaei, J. Moshfegh, A. Mohseni-Kabir, J. M. Houghton, Y. Sun, B. Kim, *RSC Adv.* **1859**, *2019*, 9.

[42] K.-M. Lee, D. Yarbrough, M. M. Kozman, T. J. Herrman, J. Park, R. Wang, D. Kurouski, *Food Control* **2020**, *114*, 107243.

[43] W. J. Thrift, R. Ragan, *Anal. Chem.* **2019**, *91*, 13337.

[44] M. Erzina, A. Trelin, O. Guselnikova, B. Dvorankova, K. Strnadova, A. Perminova, P. Ulbrich, D. Mares, V. Jerabek, R. Elashnikov, V. Svorcik, O. Lyutakov, *Sens. Actuators B* **2020**, *308*, 127660.

[45] X. Shao, H. Zhang, Y. Wang, H. Qian, Y. Zhu, B. Dong, F. Xu, N. Chen, S. Liu, J. Pan, W. Xue, *Nanomed. Nanotechnol. Biol. Med.* **2020**, *29*, 102245.

[46] J. Ding, Q. Lin, J. Zhang, G. M. Young, C. Jiang, Y. Zhong, J. Zhang, *Anal. Bioanal. Chem.* **2021**, *413*, 3801.

[47] J. Zhu, A. S. Sharma, J. Xu, Y. Xu, T. Jiao, Q. Ouyang, H. Li, Q. Chen, *Spectrochim. Acta a Mol. Biomol. Spectrosc.* **2021**, *246*, 118994.

[48] B. Üstün, W. J. Melssen, L. M. C. Buydens, *Anal. Chim. Acta* **2007**, *595*, 299.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: J. Q. Li, P. V. Dukes, W. Lee, M. Sarkis,
T. Vo-Dinh, *J Raman Spectrosc* **2022**, 1. https://doi.org/10.1002/jrs.6447