Supplemental Document

Optics Letters

Laser-based selective BTEX sensing using deep neural networks: supplement

MHANNA MHANNA,[1] MOHAMED SY,[1] AYMAN ARFAJ,[2] JOSE LLAMAS,[2] AND AAMIR FAROOQ[1,*]

[1] *King Abdullah University of Science and Technology (KAUST), Physical Science and Engineering Division, Thuwal 23955-6900, Saudi Arabia*
[2] *Saudi Aramco, Environmental Protection, 9f Al-Midra Tower, Dhahran 31311, Saudi Arabia*
[*] *Corresponding author: aamir.farooq@kaust.edu.sa*

Supplement DOI: https://doi.org/10.6084/m9.figshare.19939973

Parent Article DOI: https://doi.org/10.1364/OL.459719

Supplemental Document

Machine learning algorithms

Support vector machines (SVMs) are supervised machine learning algorithms used for classification problems [1, 2]. SVMs were originally developed by Vanpik et al. in 1995b [3] for binary classification problems based on a risk minimization principle. By the use of kernels, SVMs aim to find and/or create a hyperplane to separate various features with maximal margins in the original or a higher dimensional space [4, 1]. Recently, SVMs have been extended to SVRMs (support vector regression machines) to solve regression problems [4] by introducing the epsilon-insensitive loss function [5]. Commonly used kernels include linear, polynomial, and Gaussian radial-based functions (RBF).

Extreme gradient boosting (XGB), originally proposed by Chen et al. in 2016 [6], is an implementation of the gradient boosting decision tree model (GBDT) developed by Friedman et al. in 2001 [7]. XGB is an ensemble supervised learning algorithm based on mixing bagging and boosting weak learners [8, 9]. Traditional gradient boosting machines (GBM) iterate multiple trees (weak learners) by using only first derivative information to make the prediction. GBMs are suitable for both regression and binary classification problems [10]. Alternatively, in XGB, both first and second order derivatives are used with a regularization function to avoid overfitting.

Artificial neural networks (ANNs) are supervised learning algorithms that mimic the human brain, as they were inspired by the biological systems of neurons and synapses [11]. The ANN architecture is composed of an input layer which handles the input tensor, an output layer which handles the target tensor, and hidden layers which contain neurons [12]. Deep neural networks (DNNs) refer to deeper architectures which contain more than one hidden layer. DNN models are developed using a fully connected multilayer perceptron (MLP) associated with backpropagation algorithm [13, 14]. Three stages are involved in the back propagation of neural networks (BPNN): a feed forward to train the input tensor, calculation and backpropagation of the resulting error, and updating the weights in order to minimize the error [15]. The back propagation is achieved by using a fixed learning rate and partial derivatives to prevent under/overfitting [1]. A schematic of the DNN model used in this paper is given in Fig. S1.

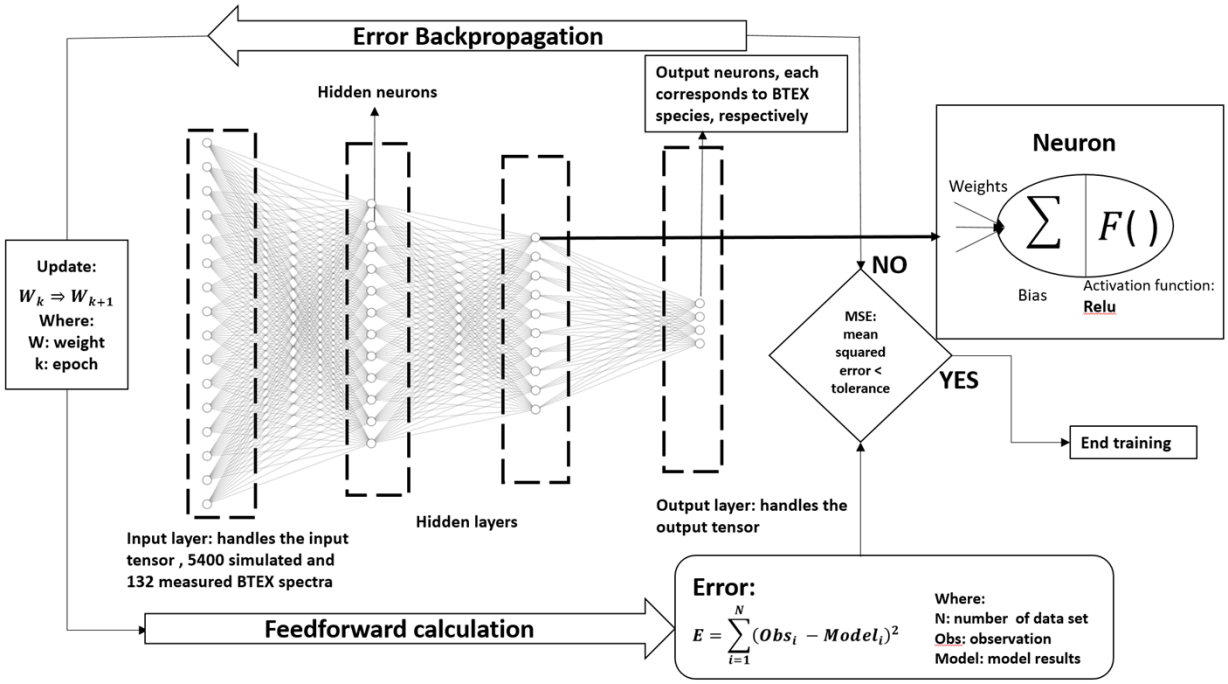

Figure S1: Schematic of the DNN model used in this work.

Cross validation:

To assess the performance and compare the algorithms, a k-fold cross validation was performed on each model. Here, the training data are divided into *k* subsets. Then, *k-1* subsets are taken as training data, and the *k*-th fold is used for validation. The process is repeated until each fold is served. Figure S2 shows the process of cross validation. In this work, a 10-fold cross validation was used while monitoring various statistical metrics: MSE (mean square error), RMSE (root mean squared error), MAE (mean absolute error), and R-squared (coefficient of determination). The equations for these metrics are given in Table S1.

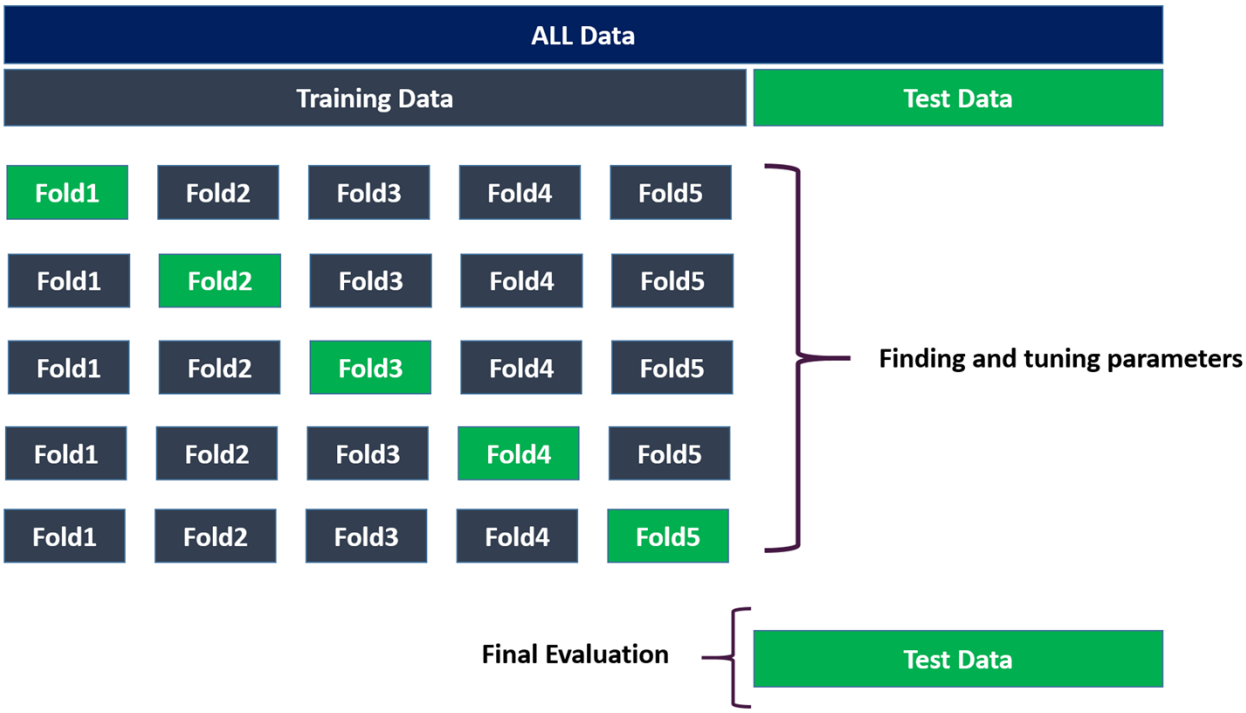

Figure S2: Cross validation schematic.

Metric	Equation
Root mean square error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - y_i)^2}{N}}$
Mean absolute error (MAE)	$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert x_i - y_i \rvert$
Coefficient of determination (R^2)	$R^2 = cor(x, y)$

Table S1: Error metrics with their corresponding equations [16].

Data Description:

A total of 5400 simulated and 132 measured BTEX absorbance spectra, over 3039.25-3040.5 cm^{-1} range, are combined. Our goal is to predict the mole fraction of each BTEX component in a BTEX mixture. The input of this model is the total BTEX absorbance, and the targets are the ratios of BTEX components. Simulated spectra were generated via a MATLAB code using Eq. S1:

$$A = \sum_{1}^{4} a_k * A_k \tag{S1}$$

where A is the composite absorption spectrum, a_k is the ratio/contribution of species k, and A_k is the normalized reference absorption spectrum of species k.

A Gaussian white noise layer was added to the training data with a standard deviation of 15% in order to mimic experimental data. Thereafter, the data are scaled using a min-max pre-processor. The goal of the scaling is to have all inputs in the same range so that the learning algorithm can predict the contributions of each species at different concentrations.

Model pathway:

Step 1: Absorbance contributions of 5400 simulated and 132 measured spectra (with known ratios) are blended in a matrix of 5532 rows and 4 columns. Each column corresponds to the ratio of one of the BTEX species. The sum of ratios in each row should be unity. Table S2 illustrates the ratio generation process.

	Benzene	Toluene	Ethylbenzene	Xylenes
row 1	0.5	0.5	0	0
row 2	0.75	0	0.25	0
row 3	0.25	0.25	0.25	0.25
row 4	0	0.2	0.6	0.2
row ...	...	...	...	...
row ...	...	...	...	...
row ...	...	...	...	...
row 5532	0.2	0.2	0.1	0.5

Table S2: Ratio generation process from simulated and measured spectra.

Step 2: After the ratio table is generated, a min-max normalization is applied to scale the data so that we only focus on contributions. The min-max normalization is given by Eq. S2:

$$A_{normalized} = \frac{A - A_{min}}{A_{max} - A_{min}} \tag{S2}$$

where $A_{normalized}$ is the normalized (scaled) absorbance, A is the composite absorbance, and A_{max} and A_{min} are the maximum and minimum absorbance values in A, respectively. Figure S3 shows the features of normalized BTEX spectra.

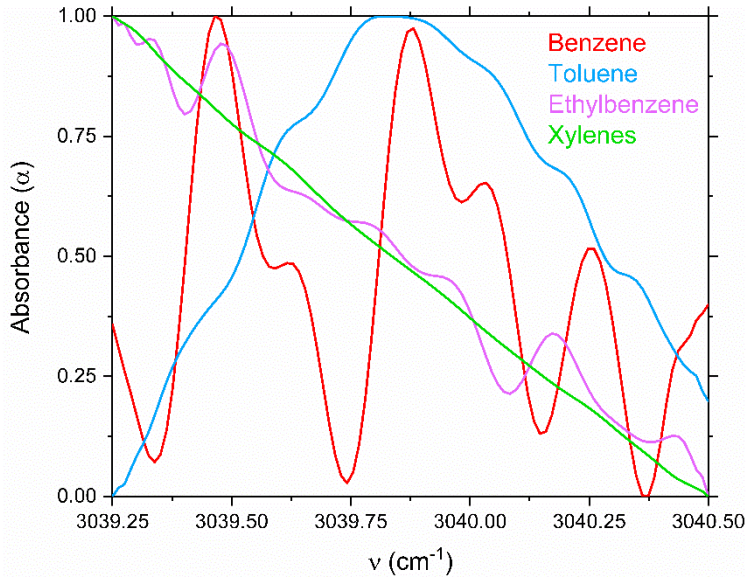

Figure S3: Normalized BTEX spectra.

Step 3: Normalized absorbance is fed to our DNN model. The input tensor is the 5532 scaled spectra and the target tensor is the ratios of each BTEX species. The data are randomly split into 80/20 train/test datasets. The model performance was monitored by computing RMSE, MAE, and R^2.

Step 4: After training the data and tuning its hyper-parameters, validation is done with new experimental data (not used in the training/testing of the model). The output of the model (ratios of all BTEX species) is used to calculate individual mole fractions using Eq. S3:

$$\chi_k = Ratio_k * \frac{A_{measured}}{A_{reference,\,k}} \tag{S3}$$

where χ_k is the mole fraction of the k-th BTEX species, $Ratio_k$ is the ratio of the k-th BTEX species obtained from the DNN model, $A_{measured}$ is the composite measured absorbance using Beer-Lambert law, and $A_{reference,\,k}$ is the reference absorbance of the k-th BTEX at T = 25 °C, P = 1 atm, L = 30 cm, and χ = 1000 ppm.

References:

[1] Al Ibrahim, Emad, and Aamir Farooq. "Octane prediction from infrared spectroscopic data." Energy & Fuels 34.1 (2019): 817-826.
[2] Sha, Huajing, et al. "A simplified HVAC energy prediction method based on degree-day." Sustainable Cities and Society 51 (2019): 101698.
[3] Cortes, C.; Vapnik, V. Support-vector networks. Machine Learning 1995, 20, 273-297.
[4] Swinehart, D. F. "The beer-lambert law." *Journal of chemical education* 39.7 (1962): 333.
[5] Louis, Bruno, Vijay K. Agrawal, and Padmakar V. Khadikar. "Prediction of intrinsic solubility of generic drugs using MLR, ANN and SVM analyses." European journal of medicinal chemistry 45.9 (2010): 4018-4025.
[6] Smola, Alex J., and Bernhard Schölkopf. "A tutorial on support vector regression." Statistics and computing 14.3 (2004): 199-222.
[7] Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system.
[8] Friedman, J.H. (2001). Greedy function approximation: A gradient boosting machine. Ann. Stat. 29: 1189–1232.
[9] Li, Wei, et al. "Gene expression value prediction based on XGBoost algorithm." Frontiers in genetics 10 (2019): 1077.
[10] Sharma, Amita, and Willem JMI Verbeke. "Improving Diagnosis of Depression with XGBOOST Machine Learning Model and a Large Biomarkers Dutch Dataset (n= 11,081)." Frontiers in Big Data 3 (2020): 15.
[11] Ma, Jinghui, et al. "Application of the XGBoost machine learning method in PM2. 5 prediction: A case study of Shanghai." Aerosol and Air Quality Research 20.1 (2020): 128-138.
[12] Agatonovic-Kustrin, S., and Rosemary Beresford. "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research." Journal of pharmaceutical and biomedical analysis 22.5 (2000): 717-727.
[13] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." Neural networks 2.5 (1989): 359-366.
[14] Nielsen, Michael A. Neural networks and deep learning. Vol. 25. San Francisco, CA: Determination press, 2015.
[15] Werbos, Paul John. The roots of backpropagation: from ordered derivatives to neural networks and political forecasting. Vol. 1. John Wiley & Sons, 1994.

[16] Guo, Hongquan, et al. "A new technique to predict fly-rock in bench blasting based on an ensemble of support vector regression and GLMNET." *Engineering with Computers* (2019): 1-15.