Machine Learning Applied for Spectra Classification

Yue Sun[1,2][0000-0001-5691-9401], Sandor Brockhauser[1,2][0000-0002-9700-4803], Péter Hegedűs[1,3][0000-0003-4592-6504]

[1] University of Szeged, Szeged, Hungary
[2] European XFEL GmbH, Schenefeld, Germany
[3] MTA-SZTE Research Group on Artificial Intelligence, ELKH, Szeged, Hungary
{yue.sun|sandor.brockhauser}@xfel.eu, hpeter@inf.u-szeged.hu

Abstract. Spectroscopy experiment techniques are widely used and produce a huge amount of data especially in facilities with very high repetition rates. In High Energy Density (HED) experiments with high-density materials, changes in pressure will cause changes in the spectral peak. Immediate feedback on the actual status (e.g. time-resolved status of the sample) would be essential to quickly judge how to proceed with the experiment. The two major spectral changes we aim to capture are either the change of intensity distribution (e.g., drop or appearance) of peaks at certain locations, or the shift of those on the spectrum.
In this work, we apply recent popular machine learning/deep learning models to HED experimental spectra data classification. The models we presented range from supervised deep neural networks (state-of-the-art LSTM-based model and Transformer-based model) to unsupervised spectral clustering algorithm. These are the common architectures for time series processing. The PCA method is used as data preprocessing for dimensionality reduction. Three different ML algorithms are evaluated and compared for the classification task. The results show that all three methods can achieve 100% classification confidence. Among them, the spectra clustering method consumes the least calculation time (0.069 s), and the transformer-based method uses the most training time (0.204 s).

Keywords: Spectral data, Classification, PCA, LSTM, Transformer, Clustering.

1 Introduction

High Energy Density (HED) scientific instrument focuses on the investigation of matter at high density, temperature, pressure, electric, and/or magnetic field [1]. In HED experiments with high-density materials, changes in pressure will cause changes in the spectral peaks (vanishing, shifting, or splitting). To evaluate the experiment status, the measured spectra need to be classified so that each class is assigned to a different state of the system under investigation. The two major spectral changes that we aim to capture in this study are

- the change of intensity distribution (e.g. drop or appearance) of peaks at certain locations, or

- the shift of those in the spectrum.

With recent developments in machine learning, data-driven machine learning /deep learning (ML/DL) methods have turned out to be very good at discovering intricate structures in high-dimensional data [2]. The ML/DL-based methods have applied broadly to a set of algorithms and techniques that train systems from raw data rather than a priori models [3], thus useful for research facilities that produce large, multidimensional datasets.

In this study, we aim to derive a statistical model for the application of HED spectra data classification. In this way, the actual status of the experiment can be fed back instantly according to the classification result, and the follow-up experiment can be better guided. We presented a simple and strong baseline range from supervised DL networks to unsupervised spectral clustering architecture for time series spectra data classification. Three commonly used ML/DL-based models are explored and evaluated on the same HED benchmark datasets, namely, the supervised LSTM-based, Transformer-based DL models and the unsupervised Spectral clustering ML algorithm. The PCA method is used here as data preprocessing for dimensionality reduction and speed up training or calculation. The experiment results show that all three methods can find a clear classification boundary and achieve 100% classification confidence. Among them, the spectra clustering method consumes the least calculation time (0.069 s). Although the data set is not clearly labeled, we use representative spectral curves as the training data set, which makes supervised DL models possible. Related work

1.1 Deep learning approaches

Deep neural networks have received an increasing amount of attention in time series analysis in recent years [4, 14]. A large variety of deep learning modeling approaches for time series analysis have been exploited for a wide range of tasks, such as forecasting, regression, and classification[5, 9, 14 15, 36]. The most common established deep learning models in this area are convolutional neural network (CNN) [13, 42, 43], recurrent neural networks (RNN) [5, 7, 8], and attention-based neural networks [10, 11, 14, 15, 16, 34]. Since CNN-based models can only learn local neighborhood features, recently, RNN-based models and attention-based models which can learn long-range dependencies are increasingly popular for learning from time series data [5].

Recurrent Approach. Two variants of the recurrent neural networks (RNN) models, Long Short Term Memory (LSTM) [6], GRU (Gated Recurrent Unit) [40], in particular, can effectively capture long term temporal dependencies, thus can work efficiently on various complex time series processing, prediction, recognition, and classification tasks [5, 7, 8, 38]. For example, in [8], Lipton et al. use clinical episodes as examples to first illustrate that LSTM has multi-label classification capabilities in multivariate time series. In the meantime, RNN-based architectures have also been used in combination with the CNN-based module to automatically extract the features and capture their long-term dependencies at the same time. The hybrid neural architectures have shown promising results for the automated analysis of time series [5, 9, 33, 38]. Lai et al. [5] proposed a Long- and Short-term Time-series network (LSTNet) framework for

multivariate time series forecasting. The method combines the strengths of CNN and RNN, can effectively extract short-term and long-term dependencies in data at the same time. In addition, they considered attention mechanism to alleviate nonseasonal time series prediction issue. Wu et al. [9] applied a convolutional recurrent neural network (CRNN) for hyperspectral data classification and achieved state-of-the-art performance. In 2017, Karim et al. [33] proposed two deep learning models for end-to-end univariate time series classification, namely LSTM RNN and ALSTM-FCN. The proposed model is an enhancement of a Fully Convolutional Network (FCN) with LSTM sub-module or attention LSTM sub-module. In 2019, the authors [35] introduced squeeze-and-excitation block to augment the FCN block, which can capture the contexture information and channel-wise dependencies, so that the model can be used for multivariate time series classification [35]. Interdonato et al. [37] proposed and end-to-end DuPLO DL architecture for the analysis of Satellite Image Time Series data. It involves branches of CNN and GRU, which can better represent remote sensing data and achieve better quantitative and qualitative classification performance.

Attention-Based Approach. Very recently, inspired by the Transformer scaling successes in NLP [10], researches have also successfully developed their Transformer-based or attention-based models for time series analysis task, such as video understanding [11], forecasting of multivariate time series data [14, 15], satellite image time series classification [12], and hyperspectral image (HSI) classification [16]. Unlike sequence-aligned models, Transformer or other attention-based models can process data sequences in more parallel and the applied attention mechanism can learn global dependencies in the sequence [4]. Ma et al. [39] first proposed a novel approach called Cross-Dimensional Self-Attention (CDSA) for the multivariate, geo-tagged time series data imputation task. The CDSA model can jointly capture the self-attention across multiple dimensions (time, location, measurement), yet in an order-independent way [39]. Garnot et al. [12] proposed a spatio-temporal classifier for automatic classification of satellite image time series, in which a Pixel-Set Encoder is used to extract spatial features, and a self-attention-based temporal encoder is used to extract temporal features. This architecture has made significant improvements in accuracy, time, and memory consumption. Rußwurm et al. [34] explored and compared several commonly used state-of-the-art deep learning mechanisms on preprocessed and raw satellite data, such as convolution, recurrence, and self-attention—for crop type identification. They pointed out that preprocessing can improve the classification performance of all models they applied, while the choice of model was less crucial [34]. Although in most cases, the attention-based architecture used for time series analysis is used as a supervised learning method, in 2020, Zerveas et al. [15] first proposes a transformer-based framework for unsupervised representation learning of multivariate time series. Even with very limited training samples, this model can still exceed the current state-of-the-art performance in the classification and regression tasks of multivariate time series, and can potentially be used for other downstream tasks, such as forecasting and missing value imputation [15].

1.2 Clustering approach

In the field of unsupervised learning, many machine learning methods for data classification have also been developed, such as k-nearest neighbor (KNN) [17], partial least-squares discrimination analysis (PLS-DA) [18, 19], support vector machine (SVM) [20], Extreme Learning Machine (ELM) [21], kernel extreme learning machine (KELM) [22]. As an unsupervised learning algorithm, clustering is one of the common nonparametric ML techniques and is widely used for exploratory data analysis [23]. Among them, spectral clustering is a clustering method that does not make assumptions about the global structure of the data [24]. It can solve very general problems like intertwined spirals and can be implemented efficiently even for large data sets [23]. For example, Jebara et al. [41] combined non-parametric spectral clustering with parametric hidden Markov models for time-series data analysis, and achieved great clustering accuracy.

In this work, we apply three commonly used machine learning/deep learning architectures to time series spectral data classification. Our proposed baseline models are based on the same PCA preprocessing process. The LSTM-based, Transformer-based and Spectral clustering network range from supervised DL neural networks to unsupervised ML algorithm are explored and evaluated on the same benchmark datasets.

2 Method

2.1 Dataset description

The spectral data used in this work was collected during the HED experiment, and it was obtained by azimuthal integration of raw X-ray diffraction images. The data set consists of 349 samples with each of 4023 features and is publicly available at https://zenodo.org/record/4424866. To show more clearly how the diffraction changes while the pressure on the sample is changing, we show one for every 10 diffractograms, as shown in **Fig. 1**. It can be clearly seen from this figure that the amplitude of spectral peaks changes (increases, decreases, vanishes) at certain locations, and the peaks also shift at 2θ-angle position, or split, or start to broaden. These changes correspond to the modification of the crystal lattice (e.g. indicating phase changes). Among them, 28 original spectra samples (the 16 marked in red belong to class label 0 and the 12 marked in blue belong to class label 1) are used as the training dataset in supervised methods. We also added 2800 simulated ones for training (by adding sufficiently small random noise, 100 simulated spectral curves can be added to each original diffractogram).

During the experiment, we should be able to track these changes and determine the actual state of the system in near real-time. Scientifically, the most relevant question is whether the phase transition in the sample has occurred. Since there is no ground truth information, in order to determine this, for supervised learning approaches, we got representative spectra measured (and simulated) at both the initial and final stages for training, which is marked in red or blue in **Fig. 1**. Based on this input, we should provide a judgment with minimum ambiguity at each point during the experiment.

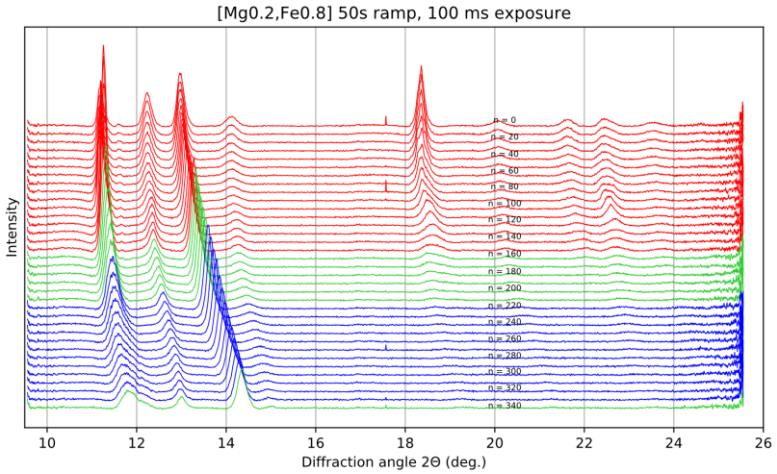

Fig. 1. Spectral data (one for every 10 diffractograms) collected during the experiment after baseline subtraction. Please note that the 28 spectra marked in red or blue are used as the basis for the LSTM or Transformer-based ML training set. Among them, the 16 marked in red belong to class label 0 and the 12 marked in blue belong to class label 1.

2.2 PCA for Dataset preprocessing

In spectroscopy experiments, it is very common that the number of input variables (features) is greater than the number of training samples, which will more easily lead to the problem of overfitting. Our data has the same characteristic. In order to facilitate the ML/DL training process, the PCA algorithm is applied to data dimensionality reduction while speeding up the training process. PCA uses an orthogonal transformation to convert data (of possibly correlated variables) into a set of new uncorrelated variables called principal components that successively maximize variance [26]. It is proved to be a simple and effective dimensionality reduction method for spectra data [22, 25].

Data centering. Before applying the PCA algorithm, the dataset features should be centered by removing the mean. Centering is performed independently on each feature by computing the relevant statistics on the samples [44], as shown in **Fig. 2**.

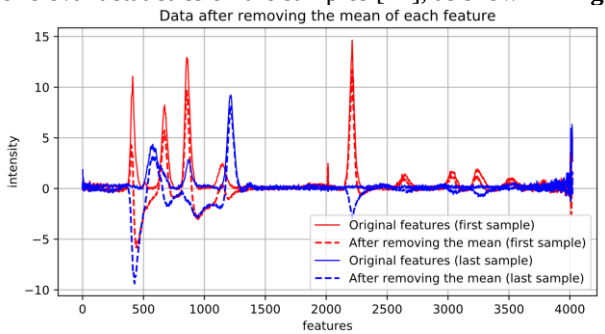

Fig. 2. Centering the dataset (take the first and last samples are used as an example).

PCA preprocessing. In the PCA method, the number of principal components (PCs) required to describe the data can be determined by looking at the cumulative explained variance ratio as a function of the number of PCs [45]. The cumulative explained variance of PCA is shown in **Fig. 3** a), the first 2 PCs explain more than 60% of the variance. Some of the new projected orthogonal variables' (PCs) values distribution can be seen from **Fig. 3** b). It can be clearly seen that the first PC explains the most variance in the data with each subsequent component explaining less.

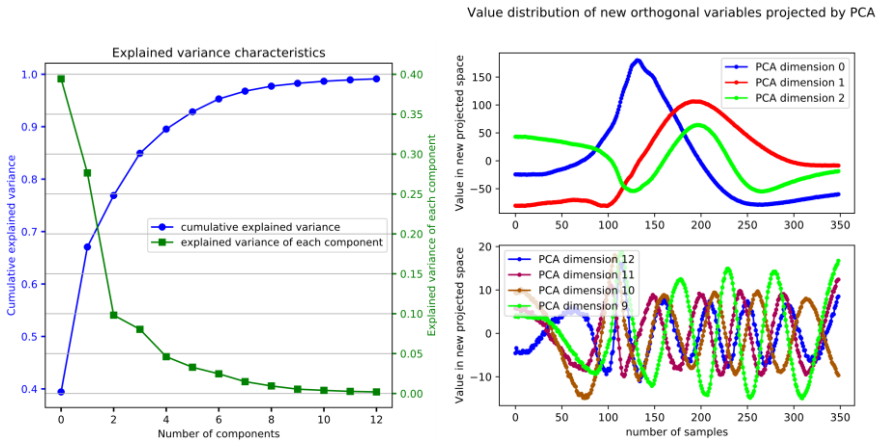

Fig. 3. a) Cumulative explained variance. b) Value distribution of new orthogonal variables projected by PCA.

When converted back to the original space, you can see the information retained or lost by the PCA algorithm more vividly, the comparison between the inverse transformation of PCA with different explained variance and the original spectra data is shown in **Fig. 4**. We can get that the first few PCs can describe the basic distribution of the data, with other PCs providing more details. In order to retain as many features as possible, we choose 13 components which can explain 99% of the variance. Then our new projected data consist of 349 samples with each of 13 features.

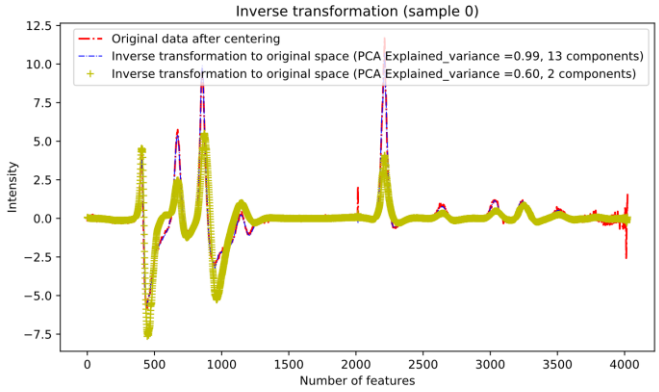

Fig. 4. Inverse transformation with different explained variances of sample 0.

Contributions of variables to PCs. In PCA, the correlation between components and variables is called loadings, it is the element of the eigenvectors and estimates the information they share [26].

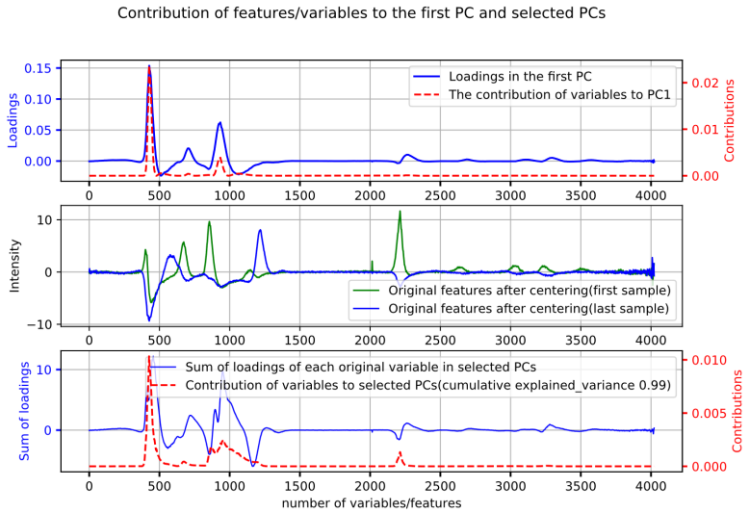

Fig. 5. Sum of loadings over variable and the contributions of each original variable/feature to the first PC and selected PCs.

The loadings (marked with blue line) and contributions (marked with red line) of variables/features in accounting for the variability to the first PC are shown in **Fig. 5** (top row). The sum of loadings and the contributions of each original variable/feature to selected PCs are shown in **Fig. 5** (bottom row). It shows that the more obvious the features/variables, the greater the contribution to the selected PCs.

2.3 LSTM-based model

As a variance of RNN in particular, Long short-term memory (LSTM), originally applied in NLP tasks, also yielded promising results for time series classification [5,34,36]. The cell unit and three gates (input gate, output gate and forget gate) in the LSTM unit allow this architecture to remember values over arbitrary time intervals and regulate the flow of information [27]. The point-wise operations used to update cell state and hidden state in the LSTM architecture can assign different weights to different features/variables in our time series spectra data, thereby improving the role of obvious features in the classification task and weakening the impact of unobvious features on classification.

Here, we do not consider the connections between different spectral observations at different time steps, but only consider the relationship between different features, that is, the sequence length is set to 1.The spectra data classification model based on LSTM structure is shown in **Fig. 6**, where the selected PCs after PCA preprocessing are fed into the LSTM unit. Here we use a single layer of LSTM cell, followed by a dense layer (64 input neurons and 1 output neuron) with Sigmoid as the activation function for the

classification task. 64 neurons are used in the hidden state. The hidden states are initialized with zero-valued vectors. The PCA preprocessing process can also be regarded as an input embedding.

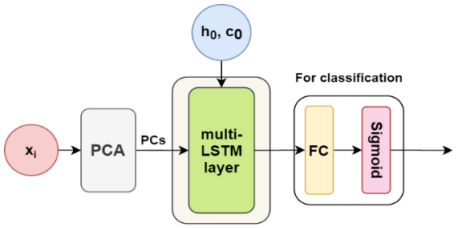

Fig. 6. Multi-LSTM layers solution for spectra data classification. PCA method is used as data preprocessing for dimensionality reduction that can also serve as an input embedding.

2.4 Transformer-based model

Transformer model relies on the so-called self-attention mechanism and is found to be superior in quality while being more parallelizable [10]. There are many successful applications of Transformers in time series processing tasks such as the spectra data classification [15, 16].

We adopted the encoder architecture of the self-attention Transformer network, as illustrated in **Fig. 7**. below. The same PCA preprocessing process is used to reduce dimensionality and save the amount of calculation. Since our spectra time series data lives in a continuous space of spectral intensity values [34], we use the dense layer or the convolutional layer for input embedding instead of a word embedding step. In addition, as with the LSTM-based method, in each batch, we only process one spectral data vector, without considering the sequential correlation of the time series, so we discarded the step of positional encoding. In this work we employed 8 attention layers, or heads, running in parallel. And the input embedding layer produces outputs of dimension 16.

In the decoder part, similarly to the input embedding, the dense layer with Sigmoid as the activation function is used to predict the class label of each spectral curve. Here, the dense layer has an input dimensionality of 16, output dimensionality of 1.

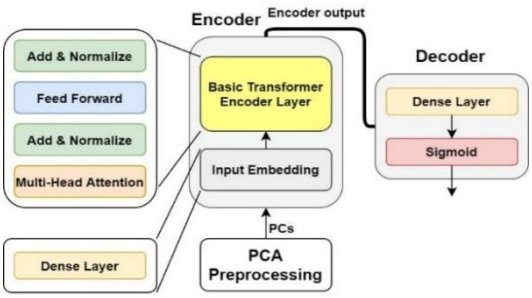

Fig. 7. Single transformer layer solution for spectra data classification. In the decoder part, the dense layer with Sigmoid as the activation function is used for the classification task.

2.5 Spectral Clustering method

Spectral Clustering uses information from the eigenvalues (spectrum) of special matrices (i.e., Affinity Matrix, Degree Matrix, and Laplacian Matrix) derived from the graph or the data set [28] and makes no assumptions about the form of the clusters. The method shows great clustering performance for data with non-convex boundaries. It is usually used when the dataset has a non-flat geometry and needs to be divided into a small number of clusters with even cluster size [30], which is well suitable for our case.

In this method, PCA preprocessing with the same parameters is used for dimensionality reduction, immediately followed by the standard spectral clustering algorithm. The clustering metrics used in the spectral clustering algorithm is graph distance, a graph of nearest neighbors [29], which is constructed to perform a low-dimension embedding of the affinity matrix between samples. And the K-Means label assignment strategy is applied in the approach, which is a popular choice [23].

2.6 Implementation

Implementation details. All the models are implemented on the Jupyter notebook platform using Pytorch and Scikit-learn libraries and use the same PCA preprocessing method with the same parameters.

The Transformer-based model and LSTM-based model are performed as supervised learning. 28 original spectra samples with 2800 simulated ones as mentioned above (by adding some random noise, 100 simulated spectral curves are generated based on each original spectrum), a total of 2828. The small random noise in simulated spectral data is generated using the Mersenne Twister [31] as the core generator. The 28 original spectra samples used as the basis for training is shown in **Fig. 1**. All training data including the simulated ones obtained after PCA preprocessing is shown in **Fig. 8**. The two models are trained by backpropagation using gradient descent, with the adaptive learning-rate method Adam [32] as the optimizer (learning rate is set to $2e^{-3}$, and weight decay is $2e^{-5}$). We use the cross-entropy loss function for our classification task. The statistical models are obtained by minimizing the loss function on the training data set. In the Transformer-based model, 15 epochs are used for iteration, and in the LSTM-based model, 45 epochs are used. The two models are trained on one machine with Tesla P100-PCIE-16GB GPU.

Jupyter notebook for reproducibility. In this work, we use Jupyter notebooks for data analysis. The analysis scripts as Jupyter notebooks are publicly available at https://github.com/sunyue-xfel/Machine-Learning-applied-for-spectra-classification.

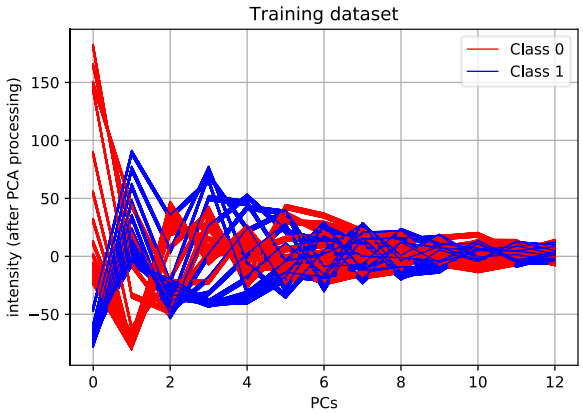

Fig. 8. Training dataset (including original spectra data and simulated ones) used in Transformer-based model and LSTM-based model.

3 Results and discussions

3.1 Performance metrics

In this study, we aim to find the phase transition point, which also means classifying the spectra into 2 phases or classes during the experiment. As there is no ground-truth phase transition information, we are interested in whether there is a clear boundary or an ambiguity zone during the experiment when the classification jumps inconsistently between the phases. Hence, our performance metric shall how small this ambiguous zone is. To explain what an ambiguity zone is, an illustrative example is shown in **Fig. 9**. Suppose we have 24 samples, corresponding to class 0 or class 1, and their classification results are shown in **Fig. 9**, the zone marked with red for the class label jump is an ambiguity zone. From the physics point of view, a proper interpretation would require the phases and the ambiguity zone to be linked to specific pressure ranges. Unfortunately, the available data is not complete and does not contain such information.

Fig. 9. An illustrative example of ambiguity zone.

Let N_f represent the number of spectral curves in ambiguous region, N_t represent the number of test spectral curves, then the classification confidence can be defined as

$$P_{conf} = 1 - \frac{N_f}{N_t} \tag{1}$$

The clear boundary between these two types of spectra yields 100% confidence. If phase transition or boundary between two classes is not detected, then all the spectral curves are in the ambiguous region, and the classification confidence is 0.

3.2 Results Comparison and discussion

Classification confidence and training time consumption of these three methods are shown in **Fig. 10**. All methods can achieve 100% classification confidence with the same PCA preprocessing process. Among them, the spectra clustering algorithm uses the least calculation time (0.069 s), and the transformer-based method consumes the most training time (0.204 s). Regarding reproducibility, all these methods have been run at least 20 times, and we get the same classification confidence and with almost the same training time, which means that they have high stability and reproducibility. Regarding complexity, for supervised learning algorithms, the parameters that need to be trained using the LSTM-based method are 20289, while the transformer-based method requires 5633. And the training losses for these two methods are 0.11917 and 0.113147, respectively.

For the spectral clustering method, we also test the classification confidence with different explained variance value which ranges from 55% to 99.99% (the corresponding number of PCs range from 2 to 301), the result shows that this method achieves consistent high-precision classification results (100% classification confidence), at the same time, the classification boundary is very stable and fluctuates only in a small range, as can be seen from **Fig. 11**. From another aspect, it also shows that the PCA algorithm can obtain the main feature information of the original data.

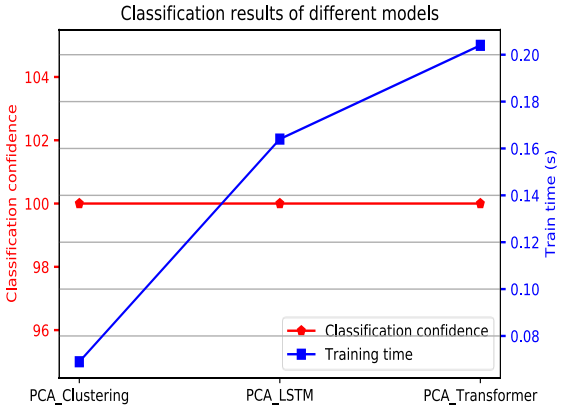

Fig. 10. Classification confidence and training time of the three models.

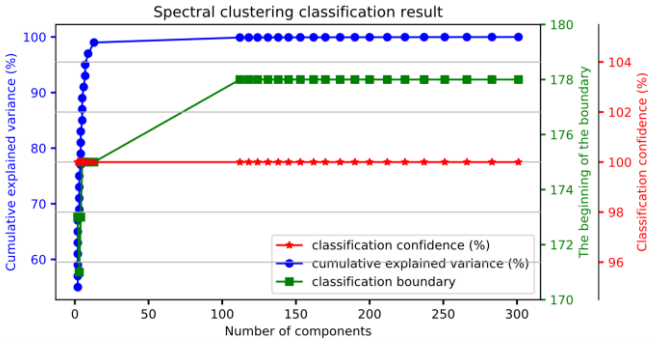

Fig. 11. Classification confidence with different explained variance value and PCs.

4 Threats to Validity

Although we obtained nice results on ML-based spectral classification, there are still some threats to validity. It can be clearly seen that in our original spectral data set, the number of training samples is limited, and the number of features is much larger than the number of samples, which will cause over-fitting problems. In this case, the PCA method is used for dimensionality reduction, and more simulated spectral curves are added to the LSTM-based and transformer-based supervisory architecture for training. However, the effect of simulation data is limited after all, and it may not reflect the real experimental data well.

In addition, since there is no ground truth information for the phase transition, the process of selecting/creating the training set is still limited. When the data is not correctly labeled and lacks some key explanatory information, we can only choose some representative spectra as training data, thus reducing the efficiency and validity of the supervised learning algorithms.

Moreover, in our current work, only one data set is used. To improve reliability and validity, multiple data sets should be used for performance evaluation and comparison.

5 Conclusion and future work

In this work, we provide a simple and strong baseline range from supervised deep neural networks to unsupervised spectral clustering architecture for time series spectra data classification. Here, the PCA method is used as data preprocessing to reduce the dimensionality and speed up the subsequent training or clustering process. The state-of-the-art supervised LSTM-based and transformer-based models are applied for spectra data classification. In these two methods, the context between different time series (sequential correlation of time series) is not considered, but only the connection between different features. Despite this, both methods achieve 100% classification confidence, a clear boundary can be found. Regarding the training time, the Transformer-based method (0.204 s) consumes more time than the LSTM-based method (0.164 s). The unsupervised spectral clustering method is also shown to be very suitable for the HED

spectra data analysis with non-flat geometries. It achieves 100% classification confidence and consumes the least amount of time (0.069 s). In addition, we provide the data analysis scripts as Jupyter notebooks for reproducibility.

In the future, for the LSTM-based and transformer-based models, we will consider using the connection between different spectral samples to better utilize the advantages of these two algorithms in time series processing. Currently, the parameters and hyperparameters of these algorithms in our work are manually selected. In subsequent research, we consider conducting parameter analysis work, for example, using some optimization algorithms to fine-tune these parameters. We will also consider applying other different deep neural network architectures, such as convolutional neural network (CNN) and its combination with LSTM or attention mechanism, to improve the model architecture of spectral classification tasks. And in future work, an end-to-end classification model without preprocessing will be introduced. Similarly, other different unsupervised clustering algorithms can be explored and compared to provide a strong baseline. At the same time, in order to better evaluate and verify the algorithm, multiple data sets from multiple experiments could be tested.

6 Acknowledgement

The authors would like to thank Christian Plueckthun and Zuzana Konopkova at European XFEL for providing the HED experimental spectral data.

This work was supported by China Scholarship Council (CSC). Furthermore, Péter Hegedűs was supported by the Bolyai János Scholarship of the Hungarian Academy of Sciences.

References

1. Nakatsutsumi, M., Tschentscher, T., Cowan, T., Ferrari, A., Schlenvoigt, H.P., Appel, K., Strempfer, J. and Zimmermann, M.V., 2014. Scientific Instrument High Energy Density Physics (HED).
2. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521: 436–444.
3. Edelen, A., Mayes, C., Bowring, D., Ratner, D., Adelmann, A., Ischebeck, R., Snuverink, J., Agapov, I., Kammering, R., Edelen, J. and Bazarov, I., 2018. Opportunities in machine learning for particle accelerators. *arXiv preprint arXiv:1811.03172*.
4. Wu, N., Green, B., Ben, X. and O'Banion, S., 2020. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*.
5. Lai, G., Chang, W.C., Yang, Y. and Liu, H., 2018, June. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (pp. 95-104).
6. S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
7. Hammerla, N.Y., Halloran, S. and Plötz, T., 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*.
8. Lipton, Z.C., Kale, D.C., Elkan, C. and Wetzel, R., 2015. Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv:1511.03677*.

9. Wu, H. and Prasad, S., 2017. Convolutional recurrent neural networks forhyperspectral data classification. *Remote Sensing*, *9*(3), p.298.
10. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.
11. Bertasius, G., Wang, H. and Torresani, L., 2021. Is Space-Time Attention All You Need for Video Understanding?. *arXiv preprint arXiv:2102.05095*.
12. Garnot, V.S.F., Landrieu, L., Giordano, S. and Chehata, N., 2020. Satellite image time series classification with pixel-set encoders and temporal self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12325-12334).
13. Wang, Z., Yan, W. and Oates, T., 2017, May. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)* (pp. 1578-1585). IEEE.
14. Shih, S.Y., Sun, F.K. and Lee, H.Y., 2019. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, *108*(8), pp.1421-1441.
15. Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A. and Eickhoff, C., 2020. A Transformer-based Framework for Multivariate Time Series Representation Learning. *arXiv preprint arXiv:2010.02803*.
16. He, X., Chen, Y. and Lin, Z., 2021. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sensing*, *13*(3), p.498.
17. Zhang, S., Li, X., Zong, M., Zhu, X. and Cheng, D., 2017. Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(3), pp.1-19.
18. R. Vitale, M. Bevilacqua, R. Bucci, A.D. Magri, A.L. Magri, F. Marini, A rapid and non-invasive method for authenticating the by NIR spectroscopy and chemometrics, Chemometr. Intell. Lab. Syst. 121 (2013) 90–99.
19. H. Chen, Z. Lin, C. Tan, Nondestructive discrimination of pharmaceutical preparations using near-infrared spectroscopy and partial least-squares discriminant analysis, Anal. Lett. 51 (2018) 564–574.
20. Zou, A.M., Shi, J., Ding, J. and Wu, F.X., 2010. Charge state determination of peptide tandem mass spectra using support vector machine (SVM). *IEEE Transactions on Information Technology in Biomedicine*, *14*(3), pp.552-558.
21. N.L. da Costa, L.A.G. Llobodanin, M.D. de Lima, I.A. Castro, R. Barbosa, Geographical recognition of Syrah wines by combining feature selection with Extreme Learning Machine, Measurement 120 (2018) 92–99.
22. Zheng, W., Shu, H., Tang, H. and Zhang, H., 2019. Spectra data classification with kernel extreme learning machine. *Chemometrics and Intelligent Laboratory Systems*, *192*, p.103815.
23. Von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and computing, 17(4), pp.395-416.
24. Jia, H., Ding, S., Xu, X. and Nie, R., 2014. The latest research progress on spectral clustering. *Neural Computing and Applications*, *24*(7), pp.1477-1486.
25. N. Tan, Y.D. Sun, X.S. Wang, A.M. Huang, B.F. Xie, Research on near infrared spectrum with principal component analysis and support vector machine for timber identification, Spectrosc. Spectr. Anal. 37 (2017) 3370–3374.
26. Jolliffe, I.T. and Cadima, J., 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), p.20150202.
27. Van Houdt, G., Mosquera, C. and Napoles, G., 2020. A review on the long short-term memory model. *Artificial Intelligence Review*, *53*, pp.5929-5955.

28. Mall, R., Langone, R. and Suykens, J.A., 2013. Kernel spectral clustering for big data networks. *Entropy*, *15*(5), pp.1567-1586.
29. White, S.; Smyth, P. A spectral clustering approach to finding communities in graphs. In Proceedings of the 2005 SIAM International Conference on Data Mining, Newport Beach, CA, USA, 21–23 April 2005; pp. 274–285.
30. Catak, F.O., Aydin, I., Elezaj, O. and Yildirim-Yayilgan, S., 2020. Practical Implementation of Privacy Preserving Clustering Methods Using a Partially Homomorphic Encryption Algorithm. *Electronics*, *9*(2), p.229.
31. Matsumoto, M. and Nishimura, T., 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *8*(1), pp.3-30.
32. Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
33. Karim, F., Majumdar, S., Darabi, H. and Chen, S., 2017. LSTM fully convolutional networks for time series classification. *IEEE access*, *6*, pp.1662-1669.
34. Rußwurm, M. and Körner, M., 2020. Self-attention for raw optical satellite time series classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, *169*, pp.421-435.
35. Karim, F., Majumdar, S., Darabi, H. and Harford, S., 2019. Multivariate LSTM-FCNs for time series classification. *Neural Networks*, *116*, pp.237-245.
36. Belagoune, S., Bali, N., Bakdi, A., Baadji, B. and Atif, K., 2021. Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems. *Measurement*, *177*, p.109330.
37. Interdonato, R., Ienco, D., Gaetano, R. and Ose, K., 2019. DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn. *ISPRS Journal of Photogrammetry and Remote Sensing*, *149*, pp.91-104.
38. Behera, R.K., Jena, M., Rath, S.K. and Misra, S., 2021. Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. *Information Processing & Management*, *58*(1), p.102435.
39. Ma, J., Shou, Z., Zareian, A., Mansour, H., Vetro, A. and Chang, S.F., 2019. CDSA: cross-dimensional self-attention for multivariate, geo-tagged time series imputation. *arXiv preprint arXiv:1905.09904*.
40. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
41. Jebara, T., Song, Y. and Thadani, K., 2007, September. Spectral clustering and embedding with hidden markov models. In *European Conference on Machine Learning* (pp. 164-175). 16Springer, Berlin, Heidelberg.
42. Abayomi-Alli, A., Abayomi-Alli, O., Vipperman, J., Odusami, M. and Misra, S., 2019, July. Multi-class classification of impulse and non-impulse sounds using deep convolutional neural network (DCNN). In *International Conference on Computational Science and Its Applications* (pp. 359-371). Springer, Cham.
43. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L. and Muller, P.A., 2019. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, *33*(4), pp.917-963.
44. Lazzeri, F., 2020. Machine Learning for Time Series Forecasting with Python. John Wiley & Sons.
45. VanderPlas, J., 2016. Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.".