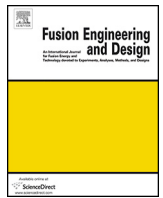




Contents lists available at [ScienceDirect](http://www.sciencedirect.com)

# Fusion Engineering and Design

journal homepage: [www.elsevier.com/locate/fusengdes](http://www.elsevier.com/locate/fusengdes)



## Automatic feature extraction in large fusion databases by using deep learning approach

Gonzalo Farias<sup>a,\*</sup>, Sebastián Dormido-Canto<sup>b</sup>, Jesús Vega<sup>c</sup>, Giuseppe Rattá<sup>c</sup>, Héctor Vargas<sup>a</sup>, Gabriel Herмосilla<sup>a</sup>, Luis Alfaro<sup>a</sup>, Agustín Valencia<sup>a</sup>

<sup>a</sup> Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

<sup>b</sup> Departamento de Informática y Automática, UNED, Madrid, Spain

<sup>c</sup> Asociación EURATOM/CIEMAT Para Fusión, CIEMAT, Madrid, Spain

### HIGHLIGHTS

- Feature extraction is a very critical stage in any machine learning algorithm.
- The problem dimensionality can be reduced enormously when selecting suitable attributes.
- Despite the importance of feature extraction, the process is commonly done manually by trial and error.
- Fortunately, recent advances in deep learning approach have proposed an encouraging way to find a good feature representation automatically.
- In this article, deep learning is applied to the TJ-II fusion database to get more robust and accurate classifiers in comparison to previous work.

### ARTICLE INFO

#### Article history:

Received 15 June 2015

Received in revised form 8 June 2016

Accepted 9 June 2016

Available online xxx

#### Keywords:

Future extraction

Machine learning

Autoencoder

Thomson scattering

### ABSTRACT

Feature extraction is one of the most important machine learning issues. Finding suitable attributes of datasets can enormously reduce the dimensionality of the input space, and from a computational point of view can help all of the following steps of pattern recognition problems, such as classification or information retrieval. However, the feature extraction step is usually performed manually. Moreover, depending on the type of data, we can face a wide range of methods to extract features. In this sense, the process to select appropriate techniques normally takes a long time. This work describes the use of recent advances in deep learning approach in order to find a good feature representation automatically. The implementation of a special neural network called sparse autoencoder and its application to two classification problems of the TJ-II fusion database is shown in detail. Results have shown that it is possible to get robust classifiers with a high successful rate, in spite of the fact that the feature space is reduced to less than 0.02% from the original one.

© 2016 Elsevier B.V. All rights reserved.

### 1. Introduction

Experiments in thermonuclear fusion generate a huge number of digital signals. Thousands of signals, sampled at high frequencies, are devoted to studying physical properties of plasma during a discharge producing massive databases.

Nowadays a shot of a tenth of a second could involve 10GB of data in some fusion devices. However, classifying an input pattern in all storage data could be very hard, and a time-consuming task, without any computer-based assistance. For that reason, a great effort has been done to apply pattern recognition and machine

learning techniques to perform pattern classification in fusion databases successfully [1–3]. However, there is still a room for improvements. In particular, feature extraction is still done manually.

Feature extraction is a very critical stage in any machine learning algorithm. Finding attributes of data that are good enough can drastically reduce the dimensionality of the problem and can make any subsequent process of classification and matching easier. On the contrary, the application of poor feature extraction methods could imply a challenge for the following stages, making it very difficult to build a successful pattern recognition system. Despite the importance of feature extraction, the process is commonly done manually by trial and error. Selection of appropriate techniques to obtain useful features from data is not straightforward and normally takes a long time. That is why any advancement in extracting

\* Corresponding author.

E-mail address: [gonzalo.farias@ucv.cl](mailto:gonzalo.farias@ucv.cl) (G. Farias).

**Table 1**  
TJ-II Thomson Scattering images.

TS Image	Description
BKG	CCD camera background.
STR	Stray light.
ECH	Electron cyclotron resonant heating.
NBI	Neutral beam injection heating.
COF	Cut-off density during electron cyclotron resonant heating.

suitable attributes automatically is very helpful. Recently, deep learning and auto encoders have emerged as one of most encouraging approaches to this end [4].

The paper is structured as follows. Section 2 introduces the datasets used in this work. In particular, images from the Thomson Scattering diagnostic and temporal evolution waveforms of a fusion device are considered. Section 3 describes the deep learning approach and in particular the attributes of autoencoders. Section 4 discusses the main results when the two datasets are classified by using extracted features with autoencoders. Finally, the main conclusions and future works are discussed.

## 2. TJ-II stellerator

In order to assess the utility of the automatic feature extraction methods for pattern recognition problems in nuclear fusion, we have selected two different classification problems from the TJ-II databases.

The TJ-II is a medium-size nuclear fusion experimental and stellerator device located at CIEMAT in Madrid (Spain). The plasmas in TJ-II are produced and heated with ECRH (2 gyrotrons, 300 kW each, 53.2 GHz, 2nd harmonic, X-mode polarization) and NBI (300 kW). In TJ-II a typical discharge lasts between 150 and 250 milliseconds, and depending on the sampling rate, the number of samples could be in the range of 4000–16000 per shot.

As mentioned before, two types of pattern recognition problems will be discussed in this work. Firstly, the classification of five types of TJ-II Thomson Scattering images, and secondly, the recognition of nine different classes of TJ-II waveforms are taken into account in order to evaluate the performance of the deep learning approach in fusion. Both problems are briefly introduced in the following subsections.

**Table 2**  
TJ-II waveforms.

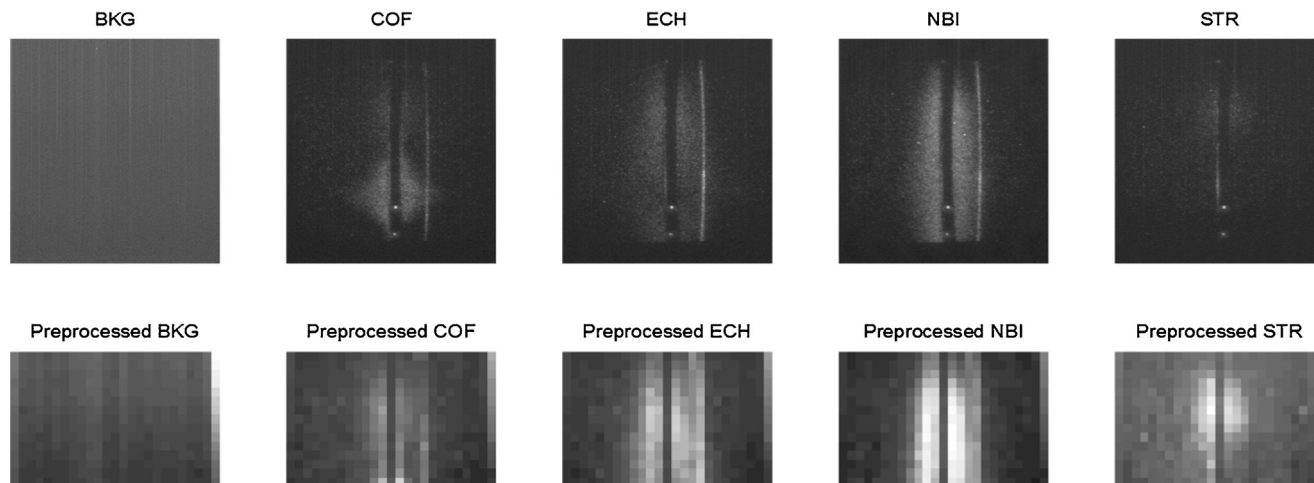
Signal	Description
ACTON275	Spectroscopic signal. CV line intensity.
BOL5	Bolometer signal. Total radiation intensity.
DENSIDAD2	Line averaged electron density.
ECE7	Electron cyclotron emission.
GR	First gyrotron.
GR2	Second gyrotron.
HALFAC3	H $\alpha$ line intensity.
IACCEL1	Neutral beam injector
RX306	Soft X-ray

### 2.1. Thomson scattering images

The Thomson Scattering (TS) diagnostic of the stellarator TJ-II provides temperature and density profiles of plasma. The diagnostic acquires five types of images (spectra of laser light scattered by plasma): CCD camera background (BKG), measurement of stray light without plasma or in a collapsed discharge (STR), images during electron cyclotron resonant heating (ECH), during neutral beam injection (NBI), and after reaching the cut-off density during electron cyclotron resonant heating (COF). Table 1 describes the five classes considered.

An automatic image classification system based on support vector machines (SVM) has been in operation for years in the TJ-II Thomson Scattering diagnostic [1,2]. SVM is a very useful method for general-purpose pattern classification. In a few words, given a set of input vectors, which belong to two different classes, SVM maps the inputs into a high-dimensional feature space through some nonlinear mapping (a kernel function), where an optimal separating hyper-plane is constructed in order to minimize the risk of misclassification. The hyper-plane is determined by a subset of points of the two classes, called **support vectors** [5]. SVM can be easily extended for a multi-class problem [2]. We have selected the TS image classification problem to show in detail preliminary results of applying the deep learning approach to image fusion databases. The experiment was carried out with 242 TS images.

Previously to extract features in an automatic way, we apply a pre-processing to all TS images. To this end we have selected Wavelet transform in order to reduce the dimensionality of the problem. According to previous works [1,2], the Wavelet parameters are defined as follows. Wavelet coefficient: approximation; Wavelet mother: Haar; Level of decomposition: 4. As a consequence of the application of Wavelet transform the image dimensions were reduced from the original  $576 \times 385$  (221760) pixels to only  $36 \times 25$



**Fig. 1.** The upper plots show the five classes of the TJ-II Thomson Scattering images. Lower plots show the corresponding pre-processed versions of the five classes.

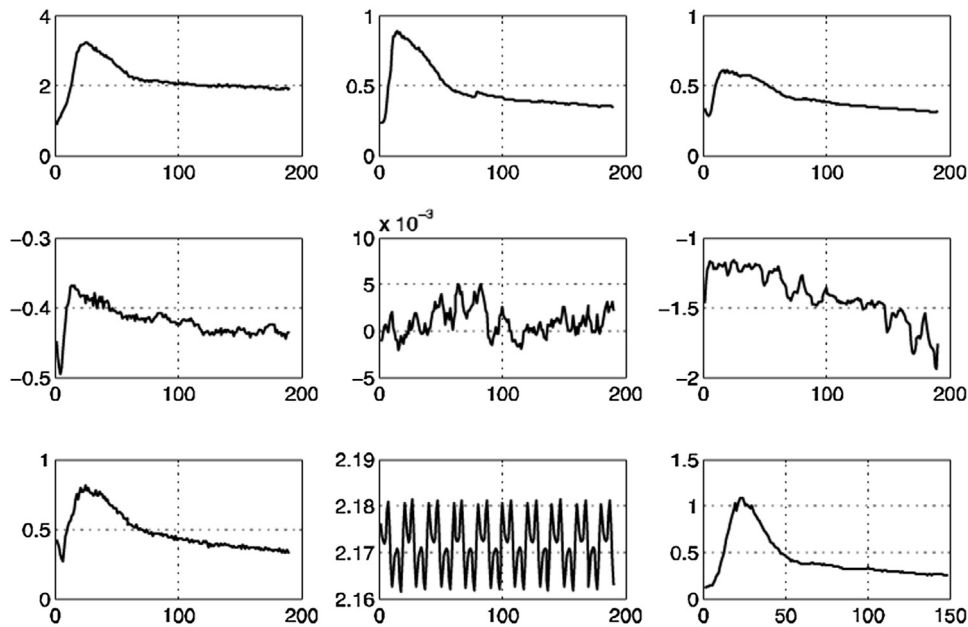


Fig. 2. TJ-II waveforms. The x-axis represents the time in milliseconds and y-axis is the amplitude of each signal, for our purposes the units are irrelevant.

(900) pixels. Finally, in order to speed-up the experimental results, we have discarded upper and lower rows of the Wavelet approximation coefficient to obtain square images of  $25 \times 25$  (625) pixels. The results of this pre-processing stage for the five types of TS images can be observed in Fig. 1.

2.2. Plasma waveforms

In [3] we have presented a pattern classification problem for recognizing temporal evolution signals from the TJ-II database. In this work, we show how the use of deep learning approach can improve the results that we have obtained before. Table 2 presents 9 different TJ-II waveforms to be considered. Fig. 2 shows the temporal evolution of these signals for a particular discharge.

Similar to the TS images, the TJ-II waveforms have also a pre-processing stage in order to reduce the problem dimensionality of the signals. In this case, we re-sampled the waveforms to 16384 samples. Wavelet transform was also applied here with an approximation coefficient, Wavelet mother Haar, and a decomposition level of 4. The application of this pre-processing stage generates signals of 1024 samples. Note that the classification was also performed by SVM technique.

3. Sparse autoencoders and deep learning

An autoencoder (AE) neural network is an unsupervised learning algorithm that applies backpropagation. In the AE we set the target values to be equal to the inputs, i.e. the auto-encoder tries to learn the identity function (see Fig. 3). This trivial function seems not to be very exciting at all, however if we consider some constraints on the AE we can discover suitable features for a learning problem in an automatic way.

For instance, a limited number of units in the hidden layer of the autoencoder should code the data in a simpler feature space. The sparseness is another interesting constraint to be considered. A sparse autoencoder is trained to keep the hidden units inactive most of the time. A unit is active when its output is close to 1, and it is inactive when its output is close to 0. Thus, we should expect that ideally only one unit is active at time in a sparse autoencoder.

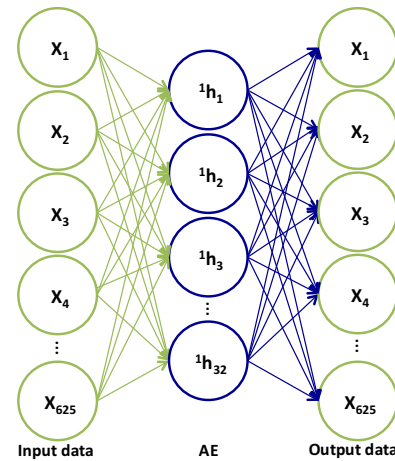


Fig. 3. An autoencoder with 625 units in the input and output layers, and with 32 units in the hidden layer.

An extended explanation of sparse autoencoders can be found in [4].

In order to test feature extraction with autoencoders we build one simple sparse autoencoder. In the case of TS images, the neural network requires one input layer of 625 units (each one represents a pixel of the input TS image), one hidden layer (the autoencoder layer), and one output layer of 625 units. The number of hidden units was modified to evaluate the performance of the classifier with a different number of features. In the case of TJ-II waveforms, the required neural network to train the AE has 1024 units in the input and output layers, as before, the hidden units are modified to evaluate the classifier.

An interesting result is to understand what the autoencoder is trying to learn in the training phase. Fortunately, we can get the input data (e.g. TS image) that activates one of the hidden units. This means that we can observe what feature has been looking for each unit of the sparse autoencoder. For instance, Fig. 4 shows the TS images that an autoencoder with four hidden units has learned to look for. Note that these images show particular features of the TS

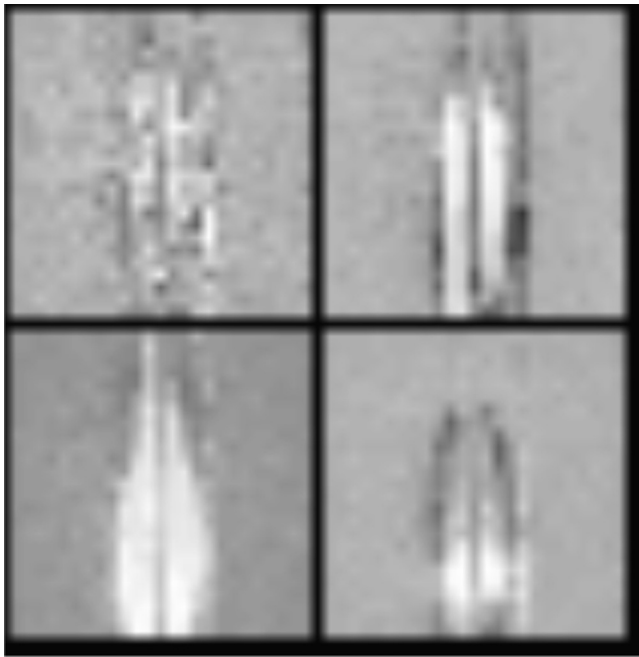


Fig. 4. Input images that activate each one of the 4 units of the autoencoder trained with the TS images.

Table 3  
 TS image classifier results with one autoencoder.

Input units	Hidden units	Successful rate (%)	Support vectors rate (%)
625	–	97.64	47.98
<b>625</b>	<b>256</b>	<b>98.57</b>	<b>15.05</b>
625	128	98.42	11.02
625	64	97.53	8.65
<b>625</b>	<b>32</b>	<b>97.41</b>	<b>6.96</b>
625	16	97.71	7.38
625	8	91.56	13.01
625	4	78.21	32.27

Note: The bold values represent the case with the highest successful rate.

image classes. In particular, features from NBI/ECH and COF images are clearly observed in the two lower input images of Fig. 4.

One advantage of this approach is the possibility to build stacked autoencoders (deep networks) in a greedy layer-wise fashion, which means that we can first train each AE separately and then use all of them together. This way allows us to avoid some well-known issues of multi-layer networks and backpropagation algorithm. Basically, such issues appear when trying to propagate back the error from the output layer to the input ones.

Fig. 5 shows a simple deep network with two autoencoders. Note that the output of the first autoencoder (AE1) feeds the second autoencoder. Moreover, the output of the second autoencoder (AE2) corresponds to the feature vector of the SVM classifier. Thus, the feature space is reduced enormously. In the case of Fig. 5, the input space starts in 625 features and it is reduced up to only 16

Table 4  
 TS image classifier results with two autoencoders.

Input units	Hidden units layer 1	Hidden units layer 2	Successful rate (%)	Support vectors rate (%)
625	256	64	97.96	7.31
625	256	32	98.49	6.63
<b>625</b>	<b>256</b>	<b>16</b>	<b>98.60</b>	<b>7.80</b>
625	256	8	97.34	11.91
625	256	4	85.57	26.11

Note: The bold values represent the case with the highest successful rate.

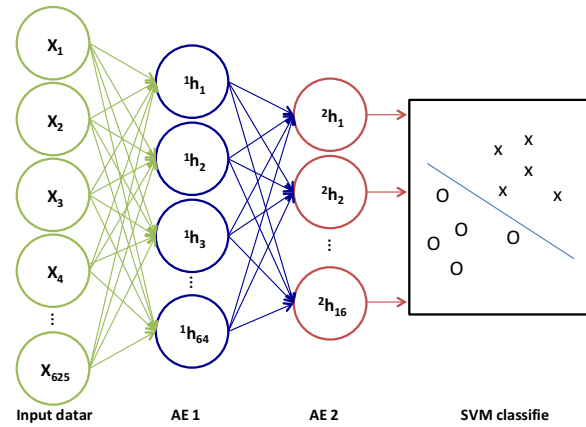


Fig. 5. Using the feature extraction with two autoencoders (i.e. a deep network) and a support vector machines to classify the TJ-II databases.

attributes. The number of units in both autoencoders is evaluated in the next section to get the best classifier.

#### 4. Experimental results

Table 3 shows the results when a simple autoencoder is used to extract features from the TS images database. As was mentioned before, the number of hidden units has been modified to assess the whole classifier. The parameters of the support vector machine are the following: kernel *Radial basis function*,  $\sigma = 8$ ,  $C = 60000$ . These parameters for the SVM classifier are constant during all the experimental evaluation. The data were divided in two datasets: a training set (which consists of 60% of the data) to build the SVM model, and a test set (which consists of 40% of the data) to evaluate the performance of the classification. Because of the training and test sets are selected randomly, the results for each experiment are the average of 100 runs.

The performance was separated based on two criteria that have been selected: The **successful rate** and the **support vector rate**. The successful rate means the percentage of hits obtained by the classifier. In our case, this value represents the average of the successful classification of all classes. Regarding the support vector rate, it represents the percentage of training data considered as support vector by the SVM model, i.e., this value is associated with the complexity of the model. Therefore, the lower the support vector rate is, the lower the risk of over-fitting exists.

Note that the first row of Table 3 represents the case without feature extraction (i.e., no AE is used in that case). Note also that the autoencoder with 32 hidden units reaches the lowest support vector rate (6,96%) and with a high successful rate (97,41%) for the TS image classifier. In the latter case, note that the feature space is reduced to less than 0.02% (32/221760) from the original one.

A second experiment was performed to evaluate a deep network with two stacked autoencoders for the TS image classifier. Results are shown in Table 4. Notice that it can be possible to have even bet-

**Table 5**  
TJ-II waveform classifier results with one autoencoder.

Input units	Hidden units	Successful rate (%)	Support vectors rate (%)
1024	–	94.68	55.08
1024	256	94.52	26.85
1024	128	94.65	15.82
1024	64	94.56	14.53
<b>1024</b>	<b>32</b>	<b>94.49</b>	<b>11.85</b>
1024	16	93.79	11.80
<b>1024</b>	<b>8</b>	<b>93.08</b>	<b>12.53</b>
1024	4	91.37	15.09
1024	2	83.37	24.92

Note: The bold values represent the case with the highest successful rate.

ter rates with a greater reduction in the feature space (see second and third rows of Table 4).

The approach has also been tested to train a classifier of nine different types of TJ-II time-series shown in Fig. 2. The experiment was carried out with 639 discharges. The methodology for training and testing was similar to the TS image classifier. Table 5 shows the results with a simple autoencoder for the nine TJ-II waveforms classifier. In this case we achieved a successful rate of 94% and less than 12% of support vectors rate, which is much better than the result we found (92% of successful rate) in previous works with only six different time-series types [3]. As before, the results show that the autoencoders help us to reduce the feature space automatically with a high performance of the developed classifiers. Note that the results are still good enough when the TJ-II waveforms classifier takes into account only eight features.

The experiments were carried out on a Mac computer using an Intel Core i7 Quad processor (2.3 GHz, 8 GB of RAM). The CPU time required for the wavelet transformation of an image or waveform is less than 15 ms. The application of the autoencoder processing required less than 1 ms in all cases. The SVM classification is less than 1 ms. This makes possible to run the classifier in about 17 ms after each shot.

## 5. Conclusions

Feature extraction is one of the most challenging issues when building learning systems. Deep learning, by means of special neural networks called autoencoders, allow us to find suitable features without human manipulation. The approach was applied to TJ-II databases with encouraging preliminary results. From the examples, we can conclude that using AE can reduce the number of original features drastically, but with a high balance of accuracy and generality.

## Acknowledgments

This work was partially supported by Chilean Ministry of Education under the Project FONDECYT 11121590 and FONDECYT 1161584. This work was partially funded by the Spanish Ministry of Economy and Competitiveness under the Projects Nos. ENE2015-64914-C3-1-R and ENE2015-64914-C3-2-R.

## References

- [1] G. Farias, R. Dormido, M. Santos, N. Duro, Image classifier for the TJ-II Thomson scattering diagnostic: evaluation with a feed forward neural network, in: *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, Springer, Berlin, Heidelberg, 2005, pp. 604–612.
- [2] L. Makili, J. Vega, S. Dormido-Canto, I. Pastor, A. Pereira, G. Farias, A. Portas, D. Pérez-Risco, M.C. Rodríguez-Fernández, P. Busch, Upgrade of the automatic analysis system in the TJ-II Thomson Scattering diagnostic: new image recognition classifier and fault condition detection, *Fusion Eng. Des.* 85 (2010) 415–418.
- [3] S. Dormido-Canto, G. Farias, R. Dormido, J. Vega, J. Sánchez, M. Santos, TJ-II waveforms analysis with wavelets and support vector machines, *Rev. Sci. Instrum.* 75 (10) (2004) 4254–4257.
- [4] Q.V. Le, Building high-level features using large scale unsupervised learning, *Acoustics, Speech and Signal Processing (ICASSP) IEEE International Conference on IEEE* (2013) 8595–8598.
- [5] V.N. Vapnik, *The Nature of Statistical Learning Theory*, second edition, Springer, Berlin, 1999, pp. 291–294.