Computer Physics Communications 320 (2026) 109984

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computational Physics

# On the use of autoencoders to study the dynamics and the causality relations of complex systems with applications to nuclear fusion☆

R. Rossi [a,*], A. Murari [b,c], T. Craciunescu [d], N. Rutigliano [a], I. Wyss [a], J. Vega [e], P. Gaudio [a], M. Gelfusa [a], on behalf of JET Contributors* and EUROfusion Tokamak Exploitation Team

[a] Department of Industrial Engineering, University of Rome "Tor Vergata", Via del Politecnico 1, Rome 00133, Italy
[b] Consorzio RFX (CNR, ENEA, INFN, University of Padova, Acciaierie Venete SpA), C.so Stati Uniti 4, Padova 35127, Italy
[c] Istituto per la Scienza e la Tecnologia dei Plasmi, CNR, Padova, Italy
[d] National Institute for Laser, Plasma and Radiation Physics, Italy
[e] Laboratorio Nacional de Fusión, CIEMAT, Madrid, Spain

ARTICLE INFO

Editor: Prof. Andrew Hazel

*Keywords:*
Autoencoders
Causality detection
Dynamical systems
Attractors
Nuclear fusion
Instabilities
ELMs
Sawteeth

ABSTRACT

Autoencoders are neural networks capable of learning compact representations of data through unsupervised learning. By encoding input data into a lower-dimensional space and subsequently reconstructing it, they enable efficient feature extraction, denoising, anomaly detection, and other applications. This work develops autoencoder-based methodologies tailored to time-dependent problems, specifically for reconstructing hidden dynamics, modelling governing equations, and detecting causal relationships.

A physics-informed autoencoder (PIC-AE) is introduced to impose physical or mathematical constraints on the latent representation, allowing the discovery of fundamental dynamics and model parameters. The PIC-AE effectively reconstructs equivalent dynamical systems from indirect measurements, as exemplified by numerical tests based on the Lotka-Volterra system of equations. It has been applied to edge-localized modes (ELMs) in nuclear fusion plasmas to assess whether they follow a Lotka-Volterra model and the results indicate the need for alternative sets of equations.

For causality detection, a novel autoencoder-based method has been developed to overcome the limitations of traditional techniques. This new approach accurately identifies causal relationships while providing a probabilistic measure of their strength. Applied to nuclear fusion data, it has confirmed the causal influence of ion cyclotron resonance heating (ICRH) on sawtooth crashes, reproducing previous findings obtained with different methodologies and extending the analysis to the spatio-temporal domain.

Although initially designed for nuclear fusion applications, the proposed methodologies are broadly applicable to any scientific and technological domain, in which time series analysis is crucial. Indeed, the developed tools have the representational capabilities of deep learning networks but are much less prone to overfitting and can be accurate even whit sparse data. Future work will explore alternative representations for ELMs and further validate the causality detection method across different datasets.

## Introduction

Sequences of experimental data acquired at subsequent points in time are called time series and constitute one of the basic forms of information analysed by scientists. Indeed, the availability of adequate time series is essential to perform various tasks such as understanding the dynamics of phenomena, predicting events or feedback control of systems [1]. Consequently, time series play a key role in several applied and technical fields, such as finance (e.g., stock market prediction and interest rate forecasting), epidemiology (e.g., disease outbreaks), energy (e.g., energy production and prediction of electricity demand), and climate science (e.g., global warming analysis and weather forecasting).

☆ See the author list of C.F. Maggi et al 2024 Nucl. Fusion 64 112012 https://doi.org/10.1088/1741-4326/ad3e16
See the author list of E. Joffrin et al 2024 Nucl. Fusion 64 112019 https://doi.org/10.1088/1741-4326/ad2be4
JET contributors §Writing – original draft, Data curation and EUROfusion Tokamak Exploitation Team£Writing – original draft, Data curation

* Corresponding author.
*E-mail address:* r.rossi@ing.uniroma2.it (R. Rossi).

https://doi.org/10.1016/j.cpc.2025.109984
Received 20 June 2025; Received in revised form 30 November 2025; Accepted 3 December 2025
Available online 4 December 2025

They are of particularly great importance in the natural sciences, where the study of time series models can help to understand phenomena that cannot be simply analysed using static (steady state) data.

The importance of time series in many fields has motivated the development of several techniques to extract information from them, such as ARIMA and SARIMA models [2], exponential smoothing [3], Vector AutoRegression (VAR) [4], Long Short-Term Memory (LSTM) networks [5], State Space Models (SSM) [6], Gaussian Processes [7], Seasonal Decomposition of Time Series (STL) [8], and Facebook Prophet [9]. In addition to time series modelling and pattern analysis, another crucial area related to time series is causality detection, which involves determining whether one quantity is influenced by one or more other variables. Causality modelling, in its turn, seeks to develop models that describe the causal relationships between time series, which is fundamental for both physical understanding and system control.

Despite the advances in various techniques for time series analysis, significant limitations remain in the task of time series modelling. Many algorithms rely on linear assumptions, while other techniques, such as LSTM networks, involve complex, non-explainable functions, which may perform well for prediction but lack interpretability for physical understanding. Furthermore, these methodologies often require substantial amounts of data, whose collection may not always be feasible given the constraints of experiments and observations. The present work therefore introduces and describes two types of tools, based on autoencoders, aimed at alleviating the aforementioned limitations and at complementing already available modelling and analysis methods. The objective consists of devising techniques with the expressive capability of deep learning neural networks, but whose outputs are interpretable and accurate even in situations of data scarcity.

The developed autoencoders therefore provide a physics-informed deep-learning framework for discovering and identifying dynamical behaviour under conditions where the true system variables are not directly observable. As such, they should be viewed as complementary to other established methods based on different technologies, such as Sparse Identification of Nonlinear Dynamical Systems (SINDy) [10,11], Dynamic Mode Decomposition (DMD)) [12,13], and Koopman theory [14,15], rather than as direct replacements for them. In any case, a detailed comparison between these methods is provided in Appendix A.

A standard autoencoder is generally made up of two deep neural networks as shown in Fig. 1 [16]. The first neural network, known as the encoder, is a function $E$ that, given the input $X$, compresses the data into a lower dimensional latent space known as the "code" $C$. The second neural network, the decoder, is a function $D$, which has to reconstruct the input of the encoder given the code. An autoencoder has different interesting features, such as the capability of automatically denoising data and of compressing the input into a smaller dimensional representation easier to analyse. When utilised to investigate time series, autoencoders are typically not applied directly to the entire time series but to time windows. Given a time series $x(t)$, the time window $X(t)$ is defined as a vector of size $W+1$, which contains a small slice of the entire time series. Time windows can be centred, backward or forward following the definitions:

$$\begin{aligned} &\textit{centred}: \; X(t_i) = \left[x\left(t_{i-W/2}\right), \ldots, x(t_i), \ldots, x\left(t_{i+W/2}\right)\right] \\ &\textit{backward}: \; X(t_i) = \left[x(t_{i-W}), \ldots, x(t_{i-1}), x(t_i)\right] \\ &\textit{forward}: \; X(t_i) = \left[x(t_i), x(t_{i+1}), \ldots, x(t_{i+W})\right] \end{aligned} \tag{1}$$

In the present work, only the backward time window is used, because the interest is in causal models that use only the past to predict or control

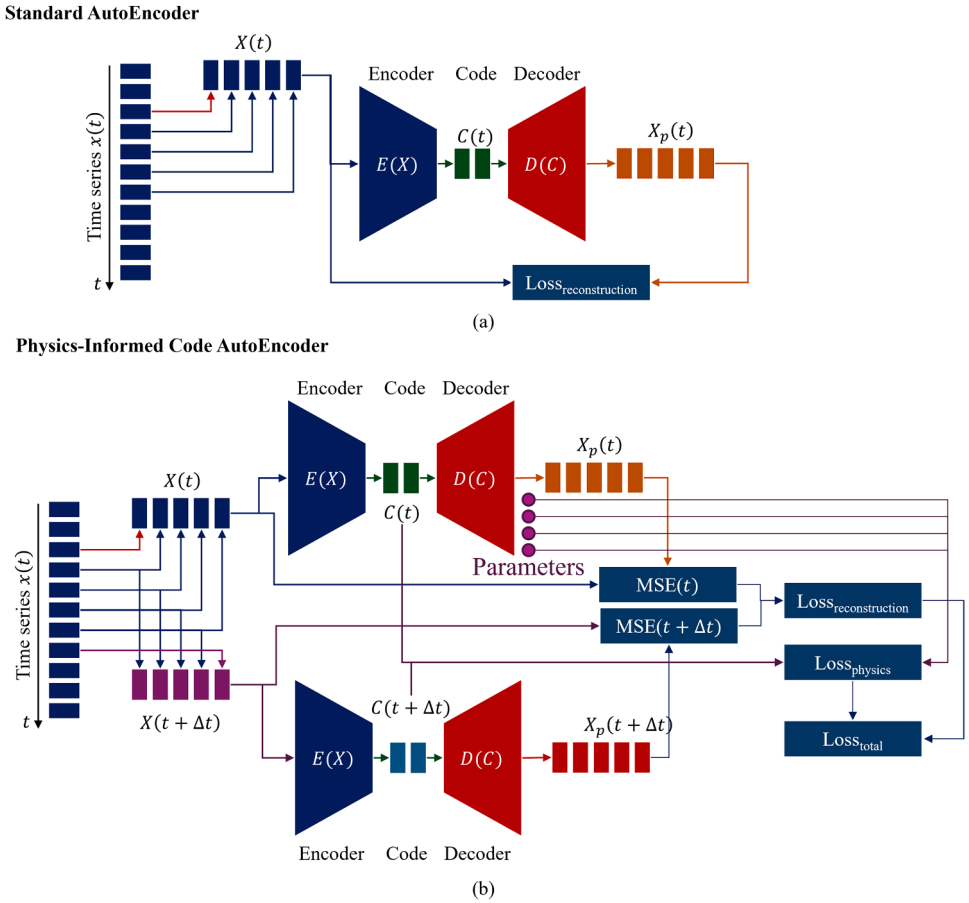


**Fig. 1.** Top: typical architecture of a traditional autoencoder. Bottom: architecture of the autoencoders developed in the framework of the present work (PIC-AE).

the future evolution of systems.

To summarise, a standard autoencoder takes as input each time window $X(t)$, compresses it into the code space $C(t)$, and then reconstructs the input time window $X_p(t)$. In general, the loss term used to train an autoencoder is the mean square error (or similar metrics) between the input $X(t)$ and the output $X_p(t)$of the autoencoder:

$$Loss_{Reconstruction} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{W+1}\sum_{j=-W/2}^{j=W/2}\left(X_p(t_{i+j}) - X(t_{i+j})\right)^2\right) \quad (2)$$

Where N is the number of time windows used in one iteration (also named "minibatch"). Sometimes, regularisation equations for the code are implemented.

This work aims to present the potential of various autoencoder architectures to tackle two main tasks: identification of system dynamics and causality detection. Indeed, one of the main lines of modern research deals with the modelling of complex behaviour with non-linear differential or finite difference equations rather than with cumbersome functions of time. Since various decades, it has become apparent that such equations even at lower order have sufficient expressive potential to describe the behaviour of many complex systems. The challenge is of course the identification of their parameters in case of noisy, incomplete and limited data. In its turn, the field of causality detection has witnessed a flurry of activity in the last decades. It has become indeed evident that quantitative information about the mutual influence between phenomena is essential not only for understanding but also for control. Consequently, directional coupling is an important output of time series analysis and modelling.

Laboratory high temperature plasmas for the investigation of thermonuclear fusion are among the most complex laboratory systems ever investigated in physics [17]. The objective consists of forcing light nuclei to coalesce into heavier ones. The challenge resides in the fact that, for the strong forces to prevail and induce the fusion, the reagents have to come very close, within distances comparable to the nucleus radius. In order for the charged particles to overcome the Coulomb repulsive barrier, they have to reach temperatures of hundred million degrees. Plasmas of these temperatures cannot come in contact with materials surfaces. The main alternative envisages their confinement with high magnetic fields in toroidal vacuum chambers. The most promising configuration of the magnetic fields is the tokamak [18].

Tokamak plasmas, particularly in reactor relevant conditions, are affected by a series of magnetic instabilities that can have a range of detrimental consequences, from the degradation of the performances to the collapse of the entire configuration [19,20]. The dynamics of many of these instabilities is not completely understood. Indeed, these phenomena, in addition to being extremely complex, involve a wide range of spatiotemporal scales. Moreover, thermonuclear plasmas are physical objects very difficult to access for measurement. Consequently, very often the quantities of interest are not available, and the modelling has to rely on proxies. These difficulties are compounded by significant electromagnetic compatibility issues, resulting in signals affected by quite high levels of noise. New methods to model the main tokamak instabilities and to clarify the causal relationships between them are in high demand.

Regarding the structure of the paper, next section presents the potential of autoencoders to identify the state variables and the attractor of complex, nonlinear and low dimensionality systems. A specific version, to integrate the available information about the structure of the desired mathematical form of the models into the autoencoder architecture, is discussed in detail. The developed tools are then deployed to model the attractor of Edge-Localised Modes (ELM) instabilities [21]. The subject of Section 3 is the description of the autoencoders potential to identify and quantify mutual causal influences between time series. Application to sawteeth pacing with ICRH modulation proves to be very interesting to determine the efficiency of this control scheme. The lines of possible future investigations are provided in the last section of the paper together with the conclusions.

## Attractor and state variables reconstruction

In many real-life applications, scientists have a limited access to the variables that can be used to describe dynamical systems in a simple and reduced form, but, on the contrary, there is access to the effects of these variables on other quantities. In this section, we introduce a version of the autoencoder technology, which integrates a physics -informed (or model-informed) code to reconstruct the dynamic of hidden variables. First, in Subsection 2.1, we present the autoencoder model called Physics-Informed Code AutoeEncoder (PIC-AE). Then, the subject of Subsection 2.2 are some synthetic tests, reported to substantiate the potential of the approach. Finally, in Subsection 2.3 the application of the autoencoder to specific instabilities in thermonuclear fusion plasmas called ELMs is tested and discussed.

### *Physics-informed code autoencoder (PIC-AE)*

A schematic of the standard autoencoder applied to time series is presented in Fig. 1(a), where it can be observed that the time windows are treated independently (the time window at time $t$ is used to predict the time window at time $t$). It has been demonstrated extensively that such an architecture possesses various interesting features to reconstruct attractors for time series [22]. However, the performances are strongly dependent on the quality of the measurements and the amount of data. Moreover, if a standard autoencoder is used, an infinite number of equivalent attractors exists and therefore each run may return a different solution.

In the present work, a training scheme and architecture, which allow to regularise the code with model equations (PDEs or others), are introduced [23,24]. The schematic of the PIC-AE is represented in Fig. 1 (b). The same encoder is used at two consecutive time instants, the inputs $X(t)$ and $X(t+\Delta t)$. The autoencoder will predict $X_p(t)$ and $X_p(t+\Delta t)$ respectively, and therefore the first loss terms are:

$$Loss_{Reconstruction} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{W+1}\sum_{j=-\frac{W}{2}}^{j=\frac{W}{2}}\left(X_p(t_{i+j}) - X(t_{i+j})\right)^2\right)$$

$$+\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{W+1}\sum_{j=-W/2}^{j=W/2}\left(X_p(t_{i+j+s}) - X(t_{i+j+s})\right)^2\right) \quad (3)$$

Where $s$ is the step size, equal to $\Delta t/\Delta t_{exp}$ where $\Delta t_{exp}$ is the experimental time resolution of the time series. Moreover, if the physics of the problem is partially known or a specific model is to be tested, the autoencoder can be trained in such a way that the code reflects specific equations. Since this aspect is case-specific, it is better discussed with a representative example by considering the Lotka-Volterra system:

$$\frac{dx}{dt} = ax - bxy$$

$$\frac{dy}{dt} = cxy - dy \quad (4)$$

To model such a system of equations, the code size of the autoencoder can be set equal to two by attributing the variable $x$ to the first code and the variable $y$ to the second.

$$x_p(t) = code_x(t);\ y_p(t) = code_y(t) \quad (5)$$

Then, by using a central finite difference scheme, one can write the Lotka-Volterra system as:

$$\frac{x_p(t+\Delta t) - x_p(t)}{\Delta t} = a x_p\left(t+\frac{\Delta t}{2}\right) - b x_p\left(t+\frac{\Delta t}{2}\right) y_p\left(t+\frac{\Delta t}{2}\right)$$

$$\frac{y_p(t+\Delta t)-y_p(t)}{\Delta t} = cx_p\left(t+\frac{\Delta t}{2}\right)y_p\left(t+\frac{\Delta t}{2}\right) - dy_p\left(t+\frac{\Delta t}{2}\right) \tag{6}$$

Where:

$$x_p\left(t+\frac{\Delta t}{2}\right) = \frac{x_p(t+\Delta t)+x_p(t)}{2}$$

$$y_p\left(t+\frac{\Delta t}{2}\right) = \frac{y_p(t+\Delta t)+y_p(t)}{2} \tag{7}$$

Therefore, one can minimise the error of fitting the two equations with the following loss term:

$$Loss_{Model} = \frac{1}{N}\sum_{i=1}^{N}\left(f_x^2(t_i)+f_y^2(t_i)\right) \tag{8}$$

Where

$$f_x = \left[x_p(t+\Delta t)-x_p(t)\right] - \left[ax_p\left(t+\frac{\Delta t}{2}\right) - bx_p\left(t+\frac{\Delta t}{2}\right)y_p\left(t+\frac{\Delta t}{2}\right)\right]\Delta t$$

$$f_y = \left[y_p(t+\Delta t)-y_p(t)\right] - \left[cx_p\left(t+\frac{\Delta t}{2}\right)y_p\left(t+\frac{\Delta t}{2}\right) - dy_p\left(t+\frac{\Delta t}{2}\right)\right]\Delta t \tag{9}$$

Finally, the total loss is evaluated as:

$$Loss_{Total} = Loss_{Reconstruction} + \alpha Loss_{Model} \tag{10}$$

Where $\alpha$ weights the importance of $Loss_{Model}$ with respect to $Loss_{Reconstruction}$. In this way, contrary to standard autoencoders, the model is trained to shape the codes in such a way that they follow the Lotka-Volterra equations. Of course, if the measurements are taken from a Lotka-Volterra system, it is reasonable to expect that both losses will be minimised. However, if the model is only one approximation of the actual system, for low $\alpha$ the reconstruction loss will be small but the model loss large. On the contrary, for large $\alpha$, the model loss would tend to zero but the autoencoder would not be able to reconstruct the input variables. This inability to find a good trade-off between the model and reconstruction loss terms would be evidence that the system can compress the data, but the assumed equations do not reflect the dynamic of the system generating the (synthetic) measurements.

The approach described above implicitly assumes that the scientist knows not only the equations of the physics, but also the parameters (in the specific case $a$, $b$, $c$ and $d$). In order to overcome this limitation, we have implemented an additional layer in the autoencoder (the layer of parameters), consisting of four isolated learnable quantities, which represent the parameters to be predicted. Therefore, the previous $f_x$ and $f_y$ are now evaluated as:

$$f_x = \left[x_p(t+\Delta t)-x_p(t)\right] - \left[a_p x_p\left(t+\frac{\Delta t}{2}\right) - b_p x_p\left(t+\frac{\Delta t}{2}\right)y_p\left(t+\frac{\Delta t}{2}\right)\right]\Delta t$$

$$f_y = \left[y_p(t+\Delta t)-y_p(t)\right] - \left[c_p x_p\left(t+\frac{\Delta t}{2}\right)y_p\left(t+\frac{\Delta t}{2}\right) - d_p y_p\left(t+\frac{\Delta t}{2}\right)\right]\Delta t \tag{11}$$

Where $a_p$, $b_p$, $c_p$ and $d_p$ are the parameters estimated by the autoencoder.

Moreover, it may happen that the scientist has access only to some of the variables of interests. It can be the case, for example, that hidden variables are accessible only with limited time-resolution or in specific intervals. In such a situation, the code may be constrained with an additional loss:

$$Loss_{Total} = Loss_{Reconstruction} + \alpha Loss_{Model} + \beta Loss_{Hidden} \tag{12}$$

Where $\beta$ is another weighting parameter and $Loss_{Hidden}$ quantifies the error between the codes and the measurements of the hidden quantities available only at certain time intervals. This loss is problem specific, but just for the sake of clarity, let us assume that we have N time intervals where the hidden variable $h$ is known. Therefore, in this specific case, the loss would be:

$$Loss_{Hidden} = \frac{1}{N}\sum_{i=1}^{N}\left(code_h(t_i) - h(t_i)\right)^2 \tag{13}$$

At this point an observation about the training and deployment of the PIC-AE architecture is probably in place. Contrary to traditional autoencoders (and to the ones used for causality detection presented in the next section), in the case of PIC-AE the dimension of the code is determined by the physics model and does not need to be found by scanning the number of its neurons. In the case of the Lotka-Volterra model just discussed, for example, the code dimension cannot be other than two. On the other hand, the weight of the physics loss α is a hyper-parameter. The appropriate value can be obtained either with a scan or in an adaptive way. In the case reported in subsection 2.3, we have implemented an adaptive scheme based on a target $Loss_{reconstruction}$, already employed in other works [25,26]. The main idea is that the reconstructed measurements should be in line with the statistics of the noise. A reconstruction error much smaller than the level of the noise is a symptom of overfitting, while a too large error indicates that the model is not able to replicate the dynamics. So, one can impose that when $Loss_{reconstruction}$ is smaller than the noise level, the value of $\alpha$ is increased. On the contrary, if the fit of the experimental data is poor, the value of $\alpha$ is to be decreased. The approach can be implemented with the following simple rule:

$$\alpha_{epoch} = \begin{cases} 1.1\ \alpha_{epoch-1} \ for\ Loss_{reconstruction} < Loss_{target} \\ 0.9\ \alpha_{epoch-1} \ for\ Loss_{reconstruction} > Loss_{target} \end{cases} \tag{14}$$

This ensures that $\alpha$ is increased to appropriately high-values and, at the same time, the signals are accurately reproduced without overfitting.

With regard to the implementation details, to obtain the results reported in this section, the following architecture and hyperparameters have been used. The autoencoder is made of a fully-connected encoder and a fully-connected decoder, consisting of 7 hidden layers with 20 neurons each. The activation functions are hyperbolic tangent for hidden layers, while linear for the output one. The algorithm to update the parameters is based on ADAM [27]. The learning rate has been set constant and equal to $10^{-3}$. The termination condition is based on the maximum number of epochs (5000).

### *Synthetic cases*

The Physics-Informed Code Autoencoder has been tested with an extended series of synthetic cases to evaluate the various features of the obtained models.

The first numerical case reported is the system of the Lotka-Volterra equations, with the model parameters equal to $a = 1.1$, $b = 0.4$, $c = 0.1$ and $d = 0.4$ and the boundary conditions $x(0) = 5$ and $y(0) = 1$. The system has been simulated with a simple backward Euler scheme with a time step equal to 1 μs and sampled every 10 ms, for a total time of 100 s. The results of the simulation for the first 50 s are shown in Fig. 2 (top).

The two variables $x$ and $y$ are supposed to be unknown, while it is assumed that two measurements are available, $p = (x+y)/2$ and $q = y(1 + 1/x)$. Random noise has been added to both synthetic measurements (different noise intensities have been studied). The time traces of p and q and the resulting attractor are shown in Fig. 2 (middle). The backward time windows from $p$ and $q$ have been computed as previously described, with a time window size equal to 50.

To analyse the synthetic data generated by this Lotka-Volterra model, a PIC-AE autoencoder has been developed. The physics model imposed is the general Lotka-Volterra set of equations and the autoencoder is allowed to adjust its parameters (see previous section for equation and training details).

Fig. 2 (bottom) shows the time traces of the codes (left) and the attractor (right). It is evident that both $code_x$ and $code_y$ replicate the behaviour of x and y, even if with a different scale ($x$ ranges from ~0.5 to

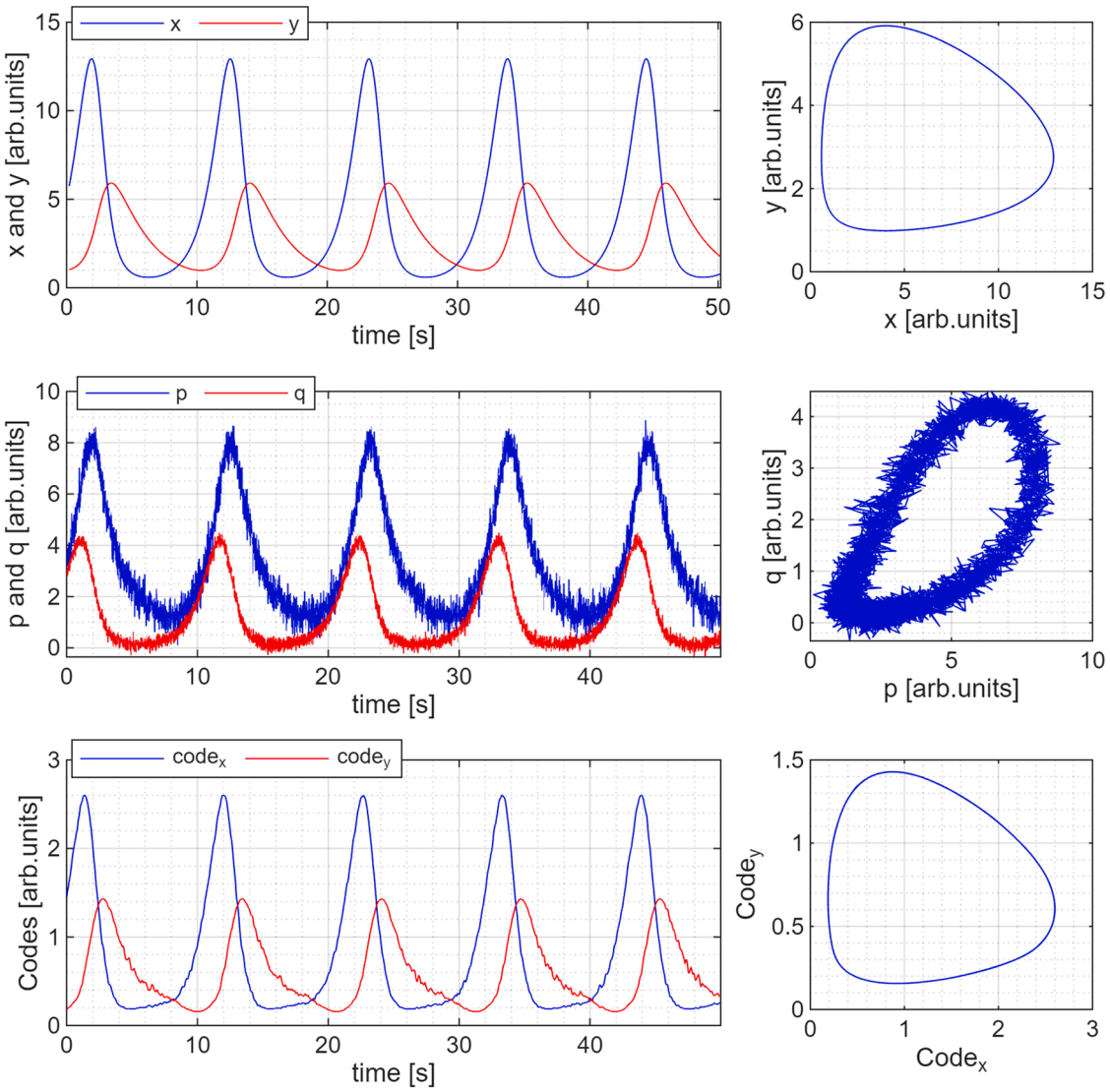


**Fig. 2.** Top: evolution of the Lotka Volterra model for parameters equal to $\boldsymbol{a} = 1.1$, $\boldsymbol{b} = 0.4$, $\boldsymbol{c} = 0.1$ and $\boldsymbol{d} = 0.4$ and boundary conditions $\boldsymbol{x}(0) = 5$ and $\boldsymbol{y}(0) = 1$. Middle: the p and q parameters after addition of noise equal to the 30 % of the average value of p and q. Bottom: reconstruction of the x and y variables obtained with the proposed autoencoder architecture.

~13 and $y$ from ~1 to ~6, while code$_x$ and code$_y$ range from ~0.2 to ~2.6 and from ~0.1 to ~1.4 respectively). However, by performing a simple linear regression analysis, one can find that the dynamic is very similar even from a quantitative point of view, with $R^2 = 99.8\%$ between $x$ and code$_x$ and $R^2 = 98.9\%$ between $y$ and code$_y$. The scatter plots between target, measured and reconstructed variables are reported in Fig. 3.

Of course, even the parameters reconstructed by the model are different from the ones used to generate the data. Indeed, while the true parameters $a$, $b$, $c$ and $d$ are 1.1, 0.4, 0.1 and 0.4, the reconstructed ones

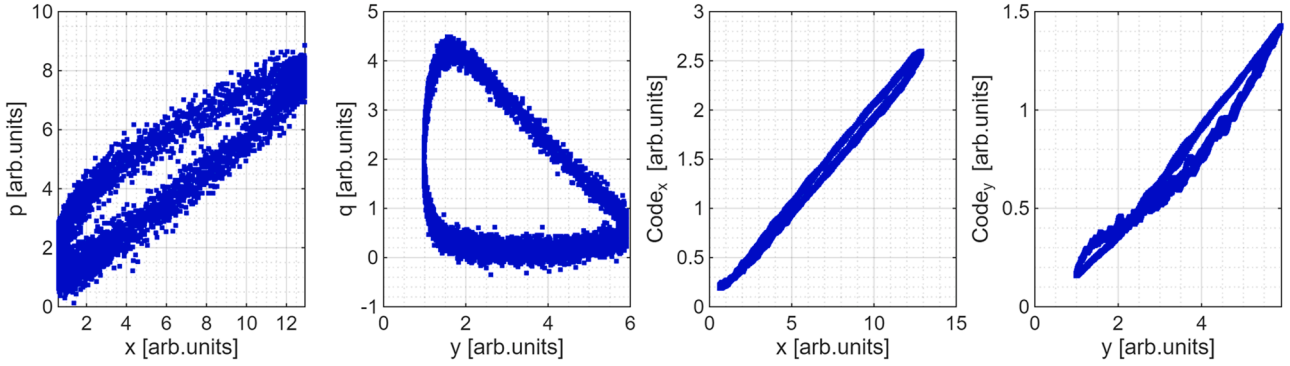


**Fig. 3.** Comparison between the target variables x and y, the measured variables p and q and the reconstructed code$_x$ and code$_y$.

($a_p$, $b_p$, $c_p$ and $d_p$) are 0.93, 1.54, 0.62 and 0.55. However, these different values are not an actual error and are due to the fact that the system is underdetermined given the proxies quantities $p$ and $q$. Consequently PIC-AE finds one of the multiple solutions that reflect exactly the system dynamics but not the original values of the model parameters. This fact can be easily proven with the help of the following considerations and equations. The PIC-AE autoencoder has identified two variables, $\text{code}_x$ ($c_x$) and $\text{code}_y$ ($c_y$), which are linearly correlated with $x$ and $y$.

$$x = m_1 c_x + q_1$$

$$y = m_2 c_y + q_2 \tag{15}$$

By substituting $x$ and $y$ inside the Lotka-Volterra equation one obtains:

$$\frac{dc_x}{dt} = ac_x + aq_1/m_1 - bm_2 c_x c_y - bm_2 c_y q_1/m_1 - bc_x q_2 - bq_1 q_2/m_1$$

$$\frac{dc_y}{dt} = cm_1 c_x c_y + cc_y q_1 + cm_1 c_x q_2/m_2 + cq_1 q_2/m_2 - dc_y - dq_2/m_2 \tag{16}$$

Which can be written as:

$$\frac{dc_x}{dt} = (a - bq_2)c_x - bm_2 c_x c_y - \frac{bm_2 c_y q_1}{m_1} - \frac{bq_1 q_2}{m_1} + \frac{aq_1}{m_1} = a_p c_x - b_p c_x c_y$$

$$\frac{dc_2}{dt} = cm_1 c_x c_y - (d - cq_1)c_y + \frac{cm_1 c_x q_2}{m_2} + \frac{cq_1 q_2}{m_2} - \frac{dq_2}{m_2} = c_p c_x c_y - d_p c_y \tag{17}$$

For the latent space to be an equivalent solution of the Lotka-Volterra system, the following equivalences must be satisfied:

$$a_p = a - bq_2$$

$$b_p = bm_2$$

$$c_p = cm_1$$

$$d_p = d - cq_1$$

$$\frac{bm_2 q_1}{m_1} = 0; -\frac{bq_1 q_2}{m_1} + \frac{aq_1}{m_1} = 0; \frac{cm_1 q_2}{m_2} = 0; \frac{cq_1 q_2}{m_2} - \frac{dq_2}{m_2} = 0 \tag{18}$$

From the previous system of equations, it is clear that there is an infinite number of correct solutions (i.e. infinite values of $m_1$, $m_2$, $q_1$ and $q_2$) to the problem. By setting $q_1$ and $q_2$ equal to zero and by measuring the two slopes $m_1$ and $m_2$, one finds that all the equalities in the last row of Eqs. (18) are satisfied, and therefore one can estimate the equivalent Lotka-Volterra parameters, which are $a = 0.93$, $b = 0.38$, $c = 0.12$ and $d = 0.55$. They are close to the real parameters but their uncertainties have to be quantified to really assess their accuracy. To this end, a Monte Carlo approach has been used to estimate the confidence intervals of the predicted parameters. By replicating the same analysis five times, we have obtained the average values and uncertainties reported in Table 1, showing that all the right values of the parameters are contained within two standard deviations of the PIC-AE autoencoder estimates. This analysis shows that the latent space of the code provides a solution equivalent to the actual Lotka-Volterra system used to generate the data. However, in a real-world scenario, the parameters $m_1$ and $m_2$ are unknown, making it impossible to recover the target parameters b and c.

**Table 1**
Target parameters vs average estimates with standard deviation.

| | a | b | c | d |
|---|---|---|---|---|
| Target | 1.10 | 0.40 | 0.10 | 0.40 |
| Average Estimate | 1.06 | 0.36 | 0.12 | 0.54 |
| Standard Deviation | 0.03 | 0.02 | 0.01 | 0.10 |

This limitation arises because, while the equations are imposed, the boundary conditions are not, leaving the problem underdetermined. The only way to fully recover the system parameters is to constrain the model using at least some direct information about x and y. This case is analysed later.

Just for comparison, in Fig. 4 the latent space reconstructed with the PIC-AE and a standard autoencoder are shown and it is clear that standard autoencoder is heavily affected by noise and that the shape of the reconstructed phase space is not a solution of the generating Lotka-Volterra system. This is of course expected, since the problem is severely ill-posed.

The same analyses have been performed varying the hyperparameters, namely minibatch size, network architecture, and external conditions, such as noise. In order to not make the paper more readable, and considering that the results are problem specific, here we just summarise the most important conclusions from the parametric study:

1. The time window size plays an important role. Too large time windows induce the autoencoders to learn global and not local features, while too small-time windows render the autoencoder unable to handle correctly noisy data. Therefore, a trade-off that must be found either by a parametric scan or by physical considerations.
2. Only a moderate sensitivity to minibatch size has been observed. The dataset size, on the contrary, is important for reliable and accurate results.
3. The model is quite robust against disturbances, with serious problems arising only for noise values over 50 % of the average signal intensity. Of course this is an aspect of great relevance in practical applications such as the experiments in thermonuclear fusion described in the next section.

The previous cases have been conducted supposing that no measurements of the variables $x$ and $y$ are possible. Of course, if one has also access to even few observations of these two quantities, one can use also these pieces of information to constrain the code (see Eq. (12)). In this situation, the code has a unique solution (if enough measurements are available) and therefore the PIC-AE will reconstruct the actual time series and should return directly the model parameters. Just to prove this, we performed another test assuming that the values of $x$ and $y$ are measurable in 10 time steps. The results are summarised in Fig. 5. With this minimal additional information, the parameters are properly identified and therefore the attractor is perfectly reconstructed.

A last analysis has been performed to demonstrate that the model does not find wrong solutions artificially, which means that if the assumed physics equations do not reflect the actual reality, the model derived by the PIC-AE will be completely different from the assumed one. In other words, the code will be projected on a latent space described by the model only if the measurements have actually been generated by the assumed model. This is important to avoid false conclusions and wrong inferences. As an example of this type of assessment, synthetic data have been generated with the Lorenz system (p and q have been evaluated as for the Lotka-Volterra case) but the Lotka-Volterra system of equations has been assumed for the physics loss term. Being the measurements simulated with a completely different physics, what we have obtained is exactly what expected. For very small values of $\alpha$, the physics is not important, and the PIC-AE just mimics a standard autoencoder behaviour. In this case, the loss on the physics is much larger than the previous examples ($\sim$10 in this case instead of $\sim$0.05). For higher values of $\alpha$, the loss physics decreases but the loss of reconstruction becomes large. For example, Fig. 6 shows the results when the reconstruction of the Lorenz system is acceptable ($R^2 \sim 95\%$). By projecting the Lorenz system on the latent space and by comparing the code with the model found by the PIC-AE, one can clearly see that the two dynamics are completely different, suggesting that the data (generated by a Lorenz system) are not compatible with the model

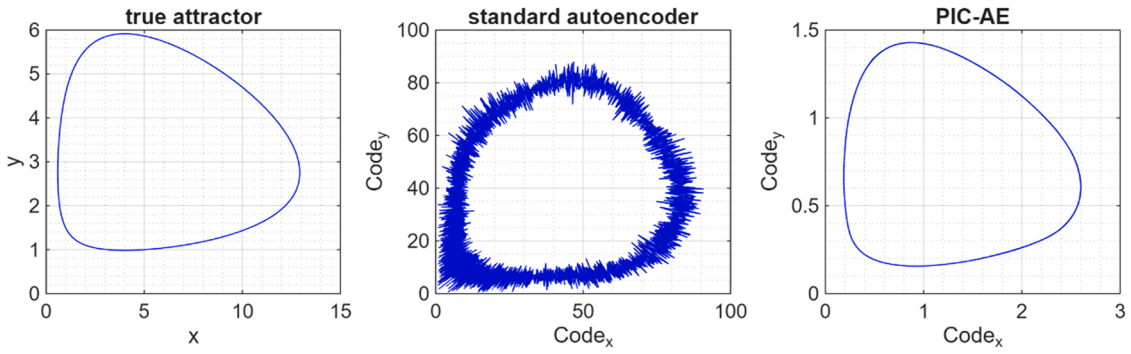


**Fig. 4.** True attractor (left), code reconstructed using a standard autoencoder (middle) and code reconstructed by the PIC-AE (right).

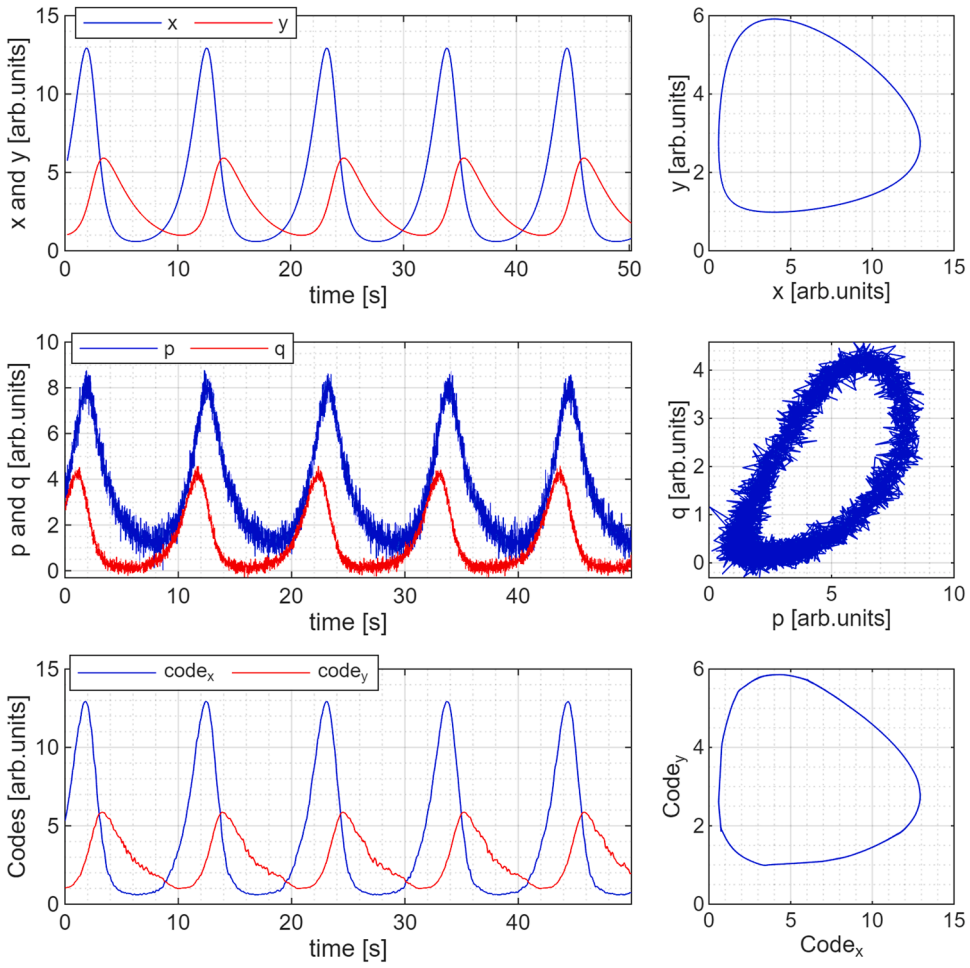


**Fig. 5.** Top: evolution of the Lotka Volterra model for parameters equal to $\boldsymbol{a} = 1.1$, $\boldsymbol{b} = 0.4$, $\boldsymbol{c} = 0.1$ and $\boldsymbol{d} = 0.4$ and boundary conditions $\boldsymbol{x}(0) = 5$ and $\boldsymbol{y}(0) = 1$. Middle: the p and q parameters after addition of noise equal to the 30 % of the average value of p and q. Bottom: reconstruction of the x and y variables obtained with the proposed autoencoder architecture, and the code constrained in ten measured points.

(Lotka-Volterra).

From the numerical analyses performed in this section together with the previous results, the following major conclusions can be drawn:

1. The implementation of a physics-informed code in the autoencoder allows projecting the data into a latent space that follows the expected physics or model. Implementing physics/model parameters as learnable parameters, the PIC-AE can be used for modelling time series.
2. However, if the measurements are generated by a system different from the assumed one, i.e. the physics/model is wrong, the autoencoder is not able to correctly compress and reconstruct the input (one of the two losses will be high depending on the value of the weight $\alpha$). This implies that the PIC-AE can also be used to test physics/model hypotheses.
3. Moreover, the PIC-AE preserves all the standard features of the autoencoders (such as denoising) but with a regularised latent space that helps improving performances.

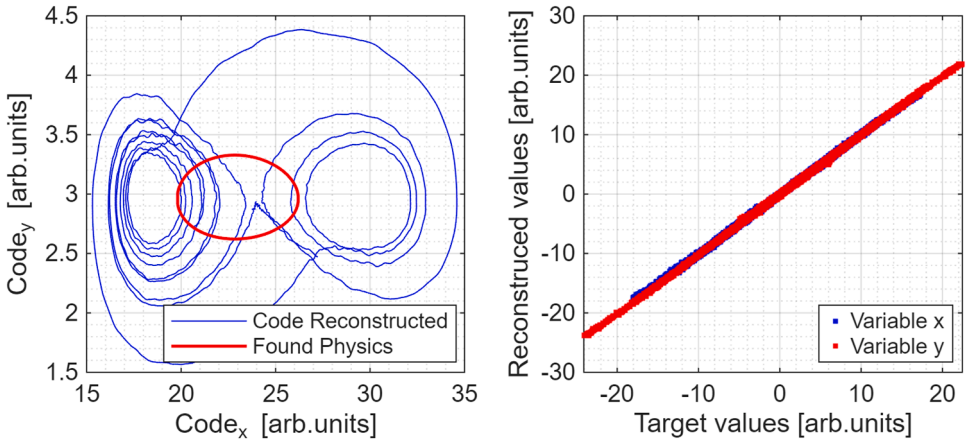


**Fig. 6.** PIC-AE with Lotka-Volterra model for the physics loss term when the data have been generated with the Lorenz system of equations.

*Application to edge localised modes*

Data in the form of time series is crucially relevant in the case of magnetically confined plasmas for the research on nuclear fusion energy via magnetic confinement [18,28]. Indeed, in these devices a large number of fast and transient phenomena are observed, such as sawtooth crashes, edge-localized modes, runaway electron (RE) beams, and other MHD instabilities [29]. Unfortunately, most of these phenomena are difficult to model, and in some cases direct measurements of the variables of interest are not available, limiting the analysis to measurements that represent only the effects, not the causes, of the phenomena. Moreover, high temperature plasmas must be actively controlled to achieve optimal performance and to avoid instabilities that can lead to disruptions, one of the main challenges for nuclear fusion tokamak reactors [30].

Edge Localised Modes (ELMs) are edge plasma instabilities that occur in high-confinement mode (H-mode) tokamak plasmas [21,31–33]. These instabilities lead to periodic bursts of energy and particles, which can cause high heat loads on plasma-facing components. ELMs have also a positive effect on the plasma, since they allow to regulate impurity levels. Understanding ELMs dynamics is important not only for better interpretation of the physics of these phenomena but also for their control, avoidance, and pacing [34,35]. One of the simplest models used to describe edge-localised modes is the Lotka-Volterra system, even if this set of equations is used more for academic purposes rather than scientific investigations [29].

The aim of this section is to show how PIC-AE can be applied to experimental measurements and to test whether the Lotka-Volterra system is actually a good candidate for describing ELMs dynamics.

In this perspective, we consider a pulse from the Joint European Torus (JET), the most successful tokamak in the world, at the moment in a decommissioning phase, after 40 years of operation [36,37]. Fig. 7

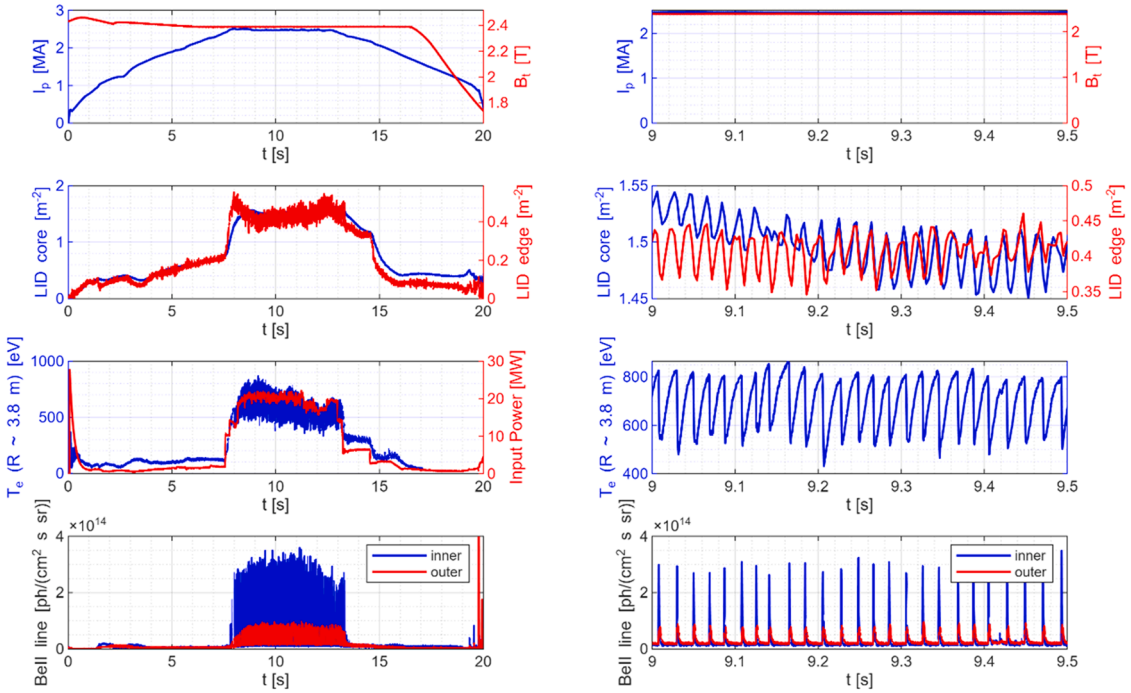


**Fig. 7.** JET Pulse 94,215. The first row shows the plasma current and the toroidal magnetic field, the second row reports the core and edge line-integrated density measured with the lines 3 and 4 of the interferometer respectively, the third row illustrates the electron temperature close to the separatrix and the total input power, while the last row shows the spectroscopic line of beryllium measured in the inner and outer part of the divertor. The left column illustrates the entire plasma discharge, while the right column reports the time window from 9.0 s to 9.5 s.

shows the main important plasma characteristics of pulse 94,217, for the entire plasma discharge (left column) and the time window from 9.0 s to 9.5 s. The first row reports the plasma current and the toroidal magnetic field, while the middle row shows the line-integrated density in the core and at the edge (line 3 and 4 of JET interferometer). The electron temperature close to the separatrix using the Electron Cyclotron Emission (ECE) diagnostic and the total input power (Ohm power and additional heatings) are shown in the third row, while the last row shows the intensity of the beryllium emission along two lines of sight, one looking at the high-field side of the divertor (inner) and another looking at the low-field side (outer). The strong and periodic bursts observed in the two Be line signals, together with the electron density and electron temperature fluctuations, are the consequences of the edge localised modes.

The analyses have been conducted in two ways. In the first case, the two BeII-lines, typically used at JET to study ELMs dynamics, are the two signals given as inputs to the PIC-AE. The second analysis utilises the electron density and electron temperature fluctuations.

In both cases, the signals have been normalised dividing the data by their standard deviation (the normalisation does not affect in any way the physics process, but it is important for the training of PIC-AE). The normalised values are indicated with the subscript $n$. Both the encoder and the decoder are two fully-connected networks with 7 layers and 20 neurons each. The code size is two, as the number of variables in the standard Lotka-Volterra system.

For the first case, a parametric analysis has been performed by varying the time window size and the physics weight $\alpha$. The step size is one, which, considering the sampling rate, corresponds to 0.1 ms. All the model parameters ($a$, $b$, $c$ and $d$) of the Lotka-Volterra system are automatically evaluated by the PIC-AE. A parametric analysis has been conducted varying the buffer size of the time-window (10, 20 and 50 points, equivalent to 1 ms, 2 ms and 5 ms respectively) and the $\alpha$ physics weight ($10^1$, $10^2$ and $10^3$). The reconstructed signals for different values of the two hyperparameters are reported against the input in Fig. 8. The plots clearly show that all the reconstructed signals are in accordance with the target, suggesting that the dimensionality of the code (two) is enough to replicate the signals in the time window considered. The $\alpha$ intensity does not influence the reconstruction capabilities (differences have not statistical significance), while the time step is slightly important, with an $R^2$ around $0.993 \pm 0.002$ for time step equal to 10, $0.986 \pm 0.004$ for time step equal to 20 and $0.982 \pm 0.002$ for time step equal to 50. However, increasing the α value to $10^4$ leads the model to collapse

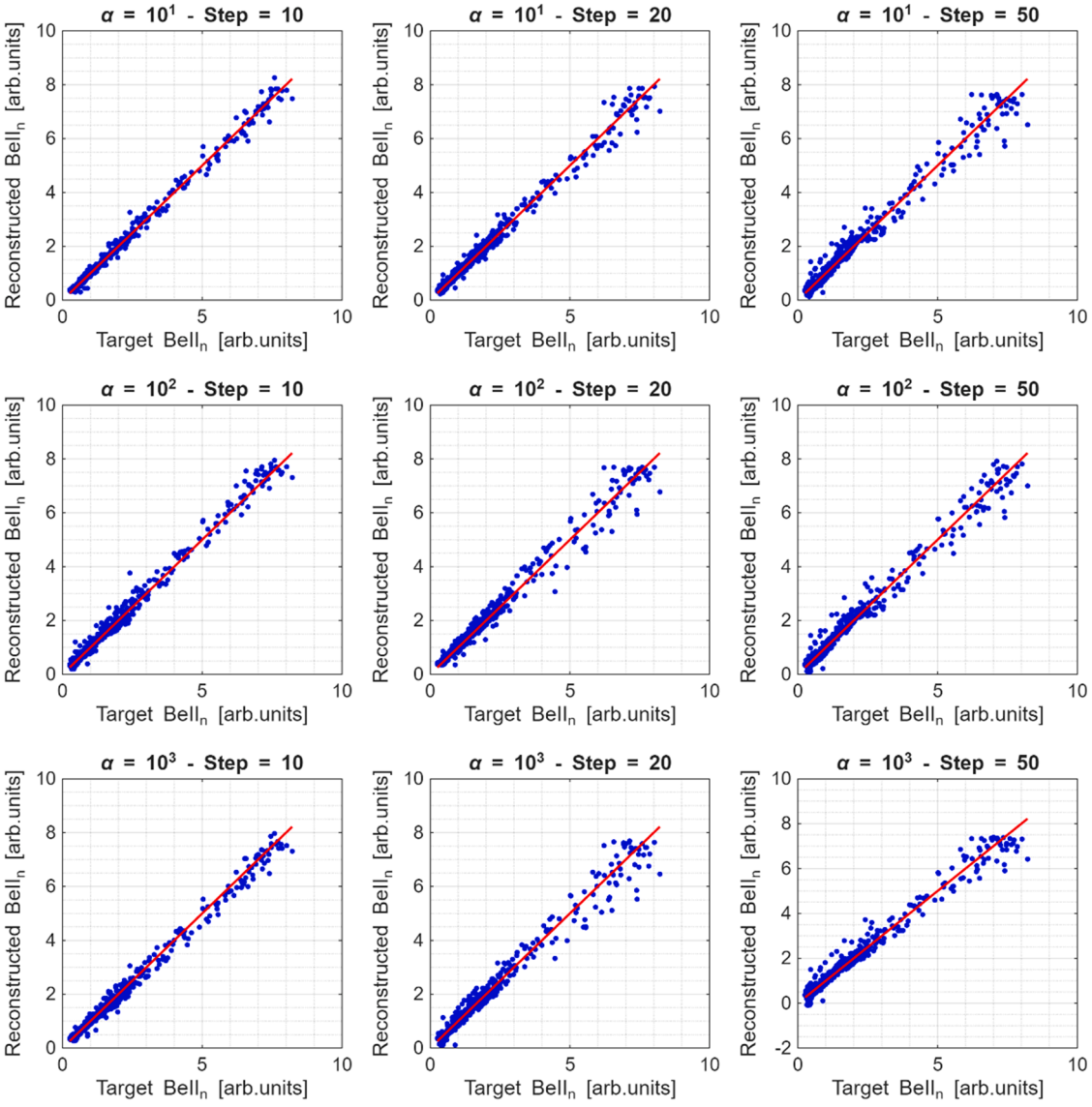


**Fig. 8.** Target vs Reconstructed $Be_n$ varying the step size and the $\alpha$ parameters. The red line is the 45 % diagonal representing the perfect agreement between the original signals and their reconstruction by the autoencoder codes.

the code to constant values and brings the reconstruction $R^2$ to zero, indicating that the physical model is not correct.

Fig. 9 compares the assumed attractors with the physical models found by the PIC-AE. The nine cases clearly show that none of them have a code projection that is close to the assumed physical model. Such a result suggests what anticipated a few lines above, i.e. that the physics model (Lotka-Volterra) does not reflect the dynamics of edge-localised modes. However, a deep analysis of ELMs dynamics, by testing other models, is outside the scope of the present work.

At last, it is worth underlining that even if the PIC-AE demonstrated that the ELM dynamics is not treatable as a Lotka-Volterra system, the denoising capabilities of the autoencoder are confirmed. This is shown in Fig. 10 for a larger and smaller time window (the PIC-AE used in this case is the one with window size of 2 ms and $\alpha$ equal to $10^2$).

In the second case, electron density and temperature fluctuations have been considered to study the ELMs dynamics. In this study, both the parametric analysis and the adaptive $\alpha$ approach have been performed, and the target loss has been evaluated as the average moving standard deviation (window size of 1 ms) of each signal. The results confirm what found in the previous cases using the BeII lines, i.e. there is not a Lotka-Volterra system that is able to correctly replicate the dynamics of ELMs. Fig. 11 shows the edge electron temperature versus the ratio between edge and core line-integrated density on the left, while the right plot shows $\text{Code}_y$ vs $\text{Code}_x$. The codes have some features (e.g. no intersections) that may help in studying and understanding the dynamics of ELMs. However, the code does not follow a Lotka-Volterra system, confirming the necessity of looking for other physics models.

### Causality detection

Up to the present, converging on a precise and practical definition of causality has been elusive even in the exact sciences. Indeed, two main lines of thought can be identified in the history of science. The first comprises the so-called deniers; the second includes those who have tried to find a single positive definition of causality [38,39]. Both approaches have proved to be clearly insufficient. The denier view of causality is clearly too simplistic, because it is now widely accepted that correlation does not imply causation. On the other hand, it has unfortunately proven impossible to agree on a positive definition of causality. The main theories claim to explain causation in terms of probability increase, energy transfer, counterfactual dependence and inferability, just to name a few [40]. Individually, they all have their merits but also flaws and often conflict with each other.

On the other hand, determining the causal relationship between time series is a fundamental task for understanding the dynamics of physical systems and developing efficient control models. Several causality

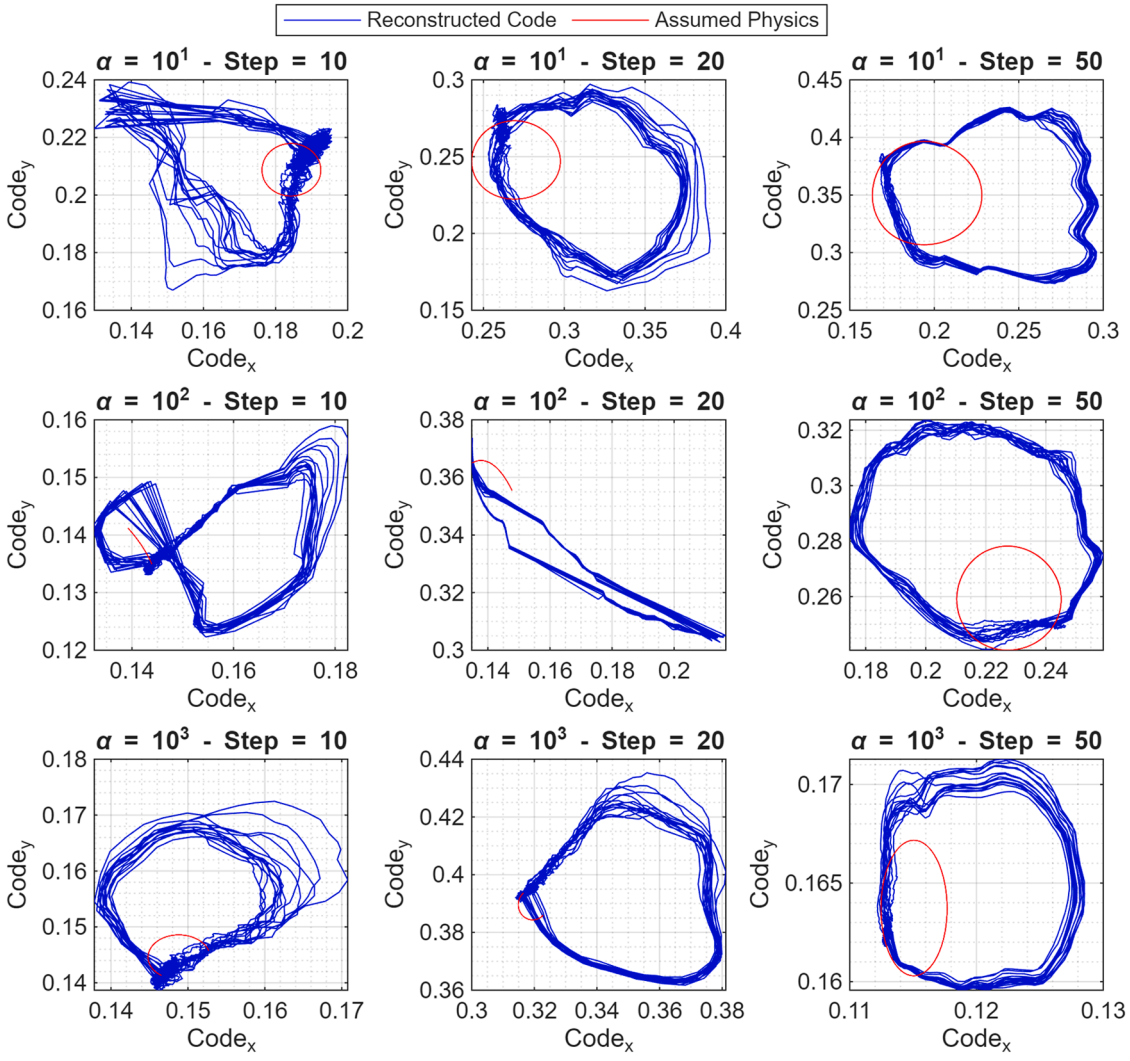


**Fig. 9.** ELM dynamics. The code space using the PIC-AE and the reconstructed physics model as a function of the two hyper-parameters. The red curves show the attractor of the Lotka-Volterra system of equations with the parameters identified by the autoencoder. The blue curve is the attractor obtained directly from the autoencoder codes.

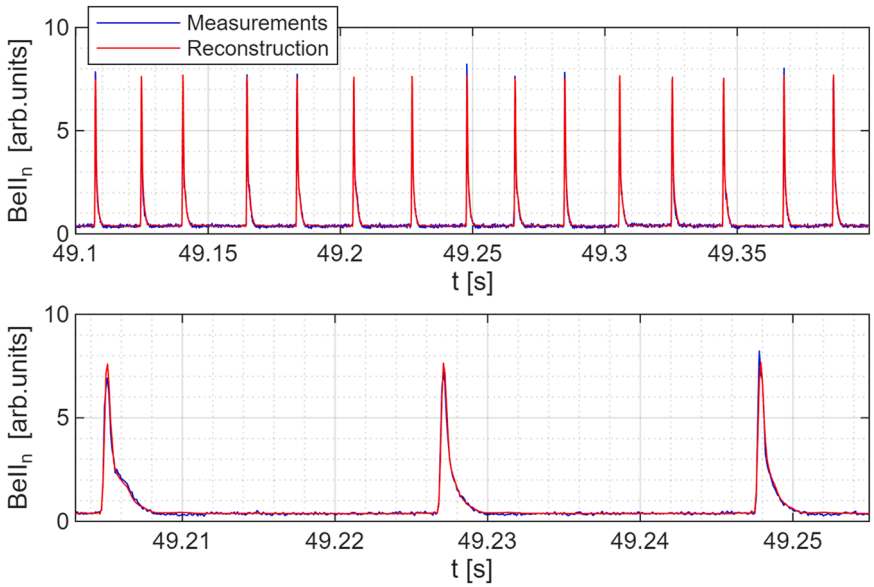


**Fig. 10.** Measured vs Reconstructed Inner Beryllium lines.

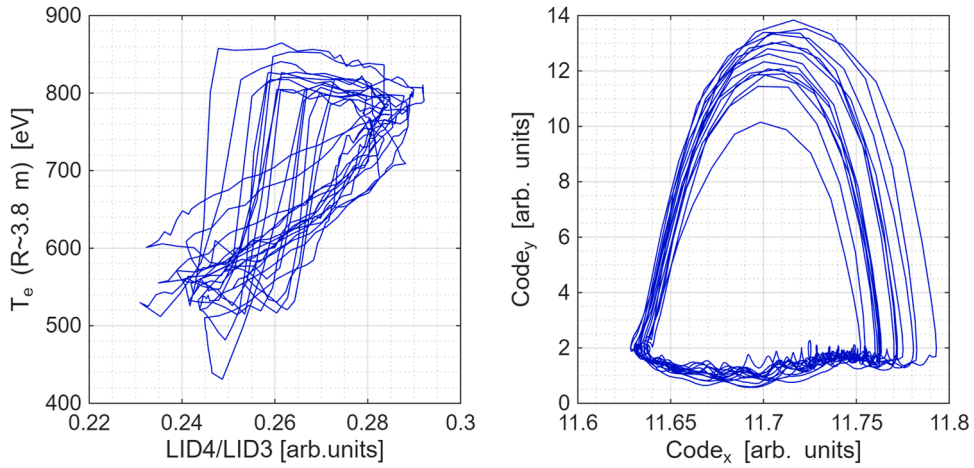


**Fig. 11.** Edge $T_e$ vs LID4/LID3 (left) and Code 1 vs Code 2 (right) using the PIC-AE with adaptive weight scheme.

detection algorithms have been tested, but various literature reviews indicate that no single technique consistently outperforms all the others. In this work the Granger Causality (GC) concept, which posits that one variable X causes another variable Y if the past values of X help in predicting Y, is assumed valid [41,42]. Various models based on Granger causality have been developed since the 60 s, such as linear Granger and Kernel Granger. However, these models have various limitations. Functionality-constrained algorithms, like linear Granger models, are limited to detecting only linear or monotone dependencies. On the other hand, deep models (such as time-delay neural networks) tend to overfit, leading to false inferences when the size of the time series is not sufficiently large. Additionally, these models require multiple hyperparameter settings, which have a very strong influence on their performances [43,44].

In this section, we present an autoencoder-based architecture to address the limitations of both approaches. Autoencoders, being deep models, can identify nonlinear functional dependencies but at the same time, reducing data to a lower-dimensional space, mitigate the risk of overfitting. The architecture of the developed autoencoders is described in detail in the next subsection. The application to a real-life experiment in the field of thermonuclear fusion is reported in Subsection 3.2, while the results of a series of systematic tests are exemplified in Appendix B.

*Causality detection with autoencoders*

In this subsection, the autoencoder architecture and causality detection methodology are presented. Let us consider three time series $x(t)$, $y(t)$ and $z(t)$ and suppose that the task at hand consists of determining whether $z(t)$ is influenced also by $x(t)$ in addition to the already known causal effect due to $y(t)$. From now on, $z(t)$ will be referred to as the "effect", while $x(t)$ as the "cause". The first step of the procedure requires extracting suitable time windows from the time series, schematically shown in Fig. 12(a), by selecting the window size $(W+1)$. From each time series, one can extract the time windows at the time $t_i$ and the time window at the next step $t_{i+s}$, where $s$ is the step. Taking the time series $x(t)$ as an example, each time window (using backward buffering) is evaluated as:

$$X(t_i) = [x(t_{i-W}), \ldots, x(t_{i-1}), x(t_i)]$$

$$X(t_{i+s}) = [x(t_{i-W+s}), \ldots, x(t_{i-1+s}), x(t_{i+s})] \tag{19}$$

Then, two autoencoders are used. The first autoencoder, shown in Fig. 12(b) and defined as $AE_{x,y,z}$, is a model that, given the time windows from all the time series at the time $t_{i_x}$, $t_{i_y}$, $t_{i_z}$, has to predict the time window of the "effect" $t_{i_z+s}$ at the next step (of course, $t_{i_x}$, $t_{i_y}$ and $t_{i_z}$ are

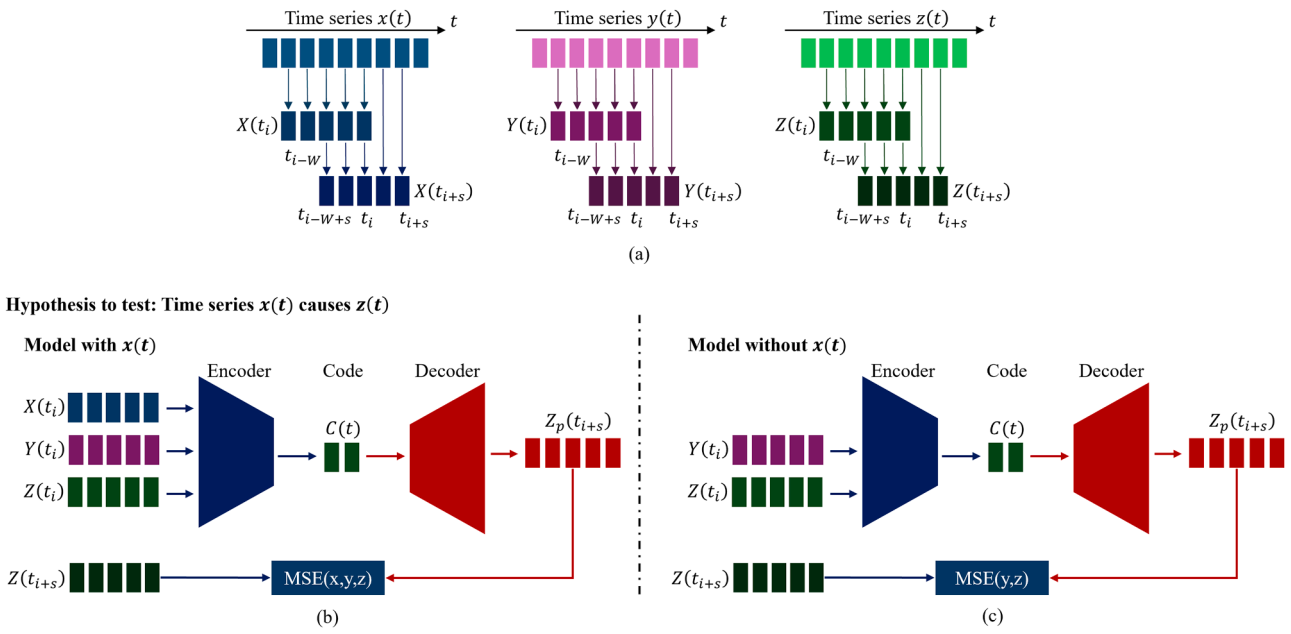


**Fig. 12.** Top: the time series and the extraction of the time windows. Bottom: the autoencoders for the determination of whether the time series x(t) influences z(t) in addition to y(t).

earlier than $t_{i_{z+s}}$). Therefore, from a mathematical point of view, the autoencoder takes as input a variable containing the three time windows:

$$I(t_i) = \left[x(t_{i_x-w}), \ldots, x(t_{i_x-1}), x(t_{i_x}), y(t_{i_y-w}), \ldots, y(t_{i_y-1}), y(t_{i_y}), z(t_{i_z-w}), \ldots, z(t_{i_z-1}), z(t_{i_z})\right] \tag{20}$$

And compresses the input into the latent space of size $M_c$. The decoder then decompresses the code to predict the time window $Z(t_{i_{z+s}})$. A second autoencoder with the same size of the latent space, shown in Fig. 12(c) and defined as $AE_{y,z}$, aims at replicating the results without including the candidate cause $x$ as information, so its input will be:

$$J(t_i) = \left[y(t_{i_y-w}), \ldots, y(t_{i_y-1}), y(t_{i_y}), z(t_{i_z-w}), \ldots, z(t_{i_z-1}), z(t_{i_z})\right] \tag{21}$$

The advantage of using this approach instead of a standard Time-Delay Neural Network (TDNN) [45,46] or other deep models is straightforward: since the model is trained to compress the data into a low dimensionality space, it will use only statistically relevant features, reducing the probability of overfitting noise or other disturbances. Obviously, this property is expected to improve the reliability of the predictions. Moreover, this architecture is not too dependent on the time window size, in the sense that one can use quite large time windows and the autoencoder will automatically tend to use only relevant information to predict the next step.

Analogously to [47], to assess the statistical relevance of the fits and their results, a hypothesis test applied to the Mean Square Errors of the residuals has been adopted. The two autoencoders are trained using the training set and the MSEs of the differences between the data and their estimates, $MSE(x,y,z)$ and $MSE(y,z)$, are evaluated for the test set. This procedure is performed several times, so that an ensemble of models is generated, which allows evaluating the average MSE and its standard deviation. Then, the statistical difference can be calculated as:

$$Z(z,x) = \frac{\mu_{MSE(y,z)} - \mu_{MSE(x,y,z)}}{\sqrt{\sigma^2_{MSE(y,z)} + \sigma^2_{MSE(x,y,z)}}} \tag{22}$$

Where $Z(z,x)$ indicates how much the knowledge of $x(t)$ is statistically important to predict the future of $z(t)$.

Concerning the model training, the time series are divided in training set and test set. Each model is trained using the training set and it is stopped using patience of 10 epochs. Contrary to the application described in Section 2, for the present task of causality detection and quantification a scan in the dimensions of the code is often necessary in practice, unless prior information is available to select this parameter.

The encoder and decoder share the same architecture as the PIC-AE: fully connected neural networks with 7 layers of 20 neurons each, using a hyperbolic tangent activation function. Parameter updates are performed with the ADAM optimizer. The maximum number of epochs is set to 5000, though early stopping consistently prevents reaching this limit. The learning rate follows a hyperbolic decay schedule, starting at $10^{-3}$ with a decay rate of $10^{-4}$.

*Experimental case: sawteeth pacing in tokamak plasmas*

The vast majority of tokamaks exhibit relaxations in the centre of the plasma column. The most distinctive signature of these instabilities is a

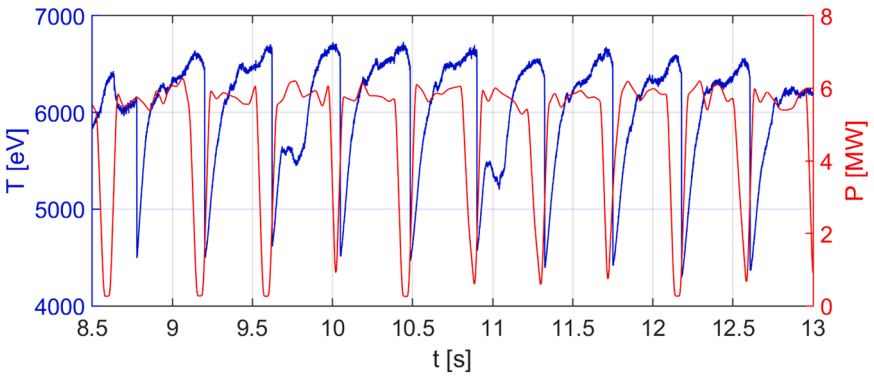


**Fig. 13.** In blue a typical time evolution of the sawteeth oscillations in the central electron temperature in JET discharge #94,089 as measured by the Electron Cyclotron Emission (ECE) diagnostic. In red the evolution of the ICRH power with the notches to destabilise the sawteeth.

non-linear oscillation of the central plasma parameters, the most affected typically being the electron temperature. These relaxation events are characterised by two different time scales: within each cycle the temperature first increases slowly and then shows a rapid crash, which normally restores the plasma conditions at the start of the cycle. The form of the oscillations resembles a sequence of sawteeth, from which they derive their name [48]. An example of these sawteeth in JET plasmas is shown in Fig. 13. The effects of the sawteeth depend on their amplitude. If they are not too large, they normally cause only moderate confinement degradation; on the other hand they could even be beneficial, by expelling impurities and helium ash and by counteracting the core accumulation of heavy impurities in devices with a metallic first wall [49–51]. On the contrary, if the amplitude of the sawteeth crash becomes excessively large, they compromise the energy confinement and can trigger various other instabilities, sometimes resulting in the total collapse of the plasma column. Consequently in the reactor it will be probably important to carefully control sawteeth. A possible alternative to influence the sawteeth behaviour consists of acting on their frequency. Indeed, given the charge and fire nature of these instabilities, increasing the frequency of the crashes tends also to reduce their amplitude.

Different methods have been envisaged and tested to control the sawtooth period in tokamaks [52]. An important alternative recently investigated on JET is based on electromagnetic waves in the radio-frequency range. These waves can indeed propagate in magnetised multi-species plasmas, which present various resonant frequencies, resulting in strong absorption and consequent significant local heating. Indeed, this so-called Ion Cyclotron Radiofrequency Heating (ICRH) can be used to transfer energy directly to the minority ions and therefore it has been deployed in the past also to investigate the physics of fast particles [53]. Modulating the ICRH power has been proposed as a technique to control the sawteeth frequency, what is called sawteeth pacing [54,55]. The mechanism, by which ICRH influences the frequency of the sawteeth, is believed to be its effect on the pressure and distribution function of energetic ions in the plasma. In their turn, the fast ions are considered to exert a stabilising effect on the instability. Therefore the sawteeth pacing scheme, implemented in the experiments analysed in this subsection, consists of modulating the central ICRH power. Increasing the ICRH power stabilises the $m = 1$ mode, resulting in longer sawteeth periods; then rapidly switching this power off it is expected to suddenly induce the crash. Present research is focussed on quantifying the strength of the ICRH perturbation necessary for successful sawteeth pacing.

Leaving aside the technical challenges, which have been brilliantly overcome on JET [51,54,56,57], a typical difficulty of these experiments is their interpretation. Indeed, from an operational perspective it is necessary to quantify reliably the efficiency of the pacing, i.e. the number of sawteeth effectively triggered by the external modulation. This is not a simple task, as can be appreciated by inspection of the typical time series reported in Fig. 13. The main difficulty resides in the fact that sawteeth are quasiperiodic and therefore a crash would always occur after a notch in the ICRH power due to the natural plasma dynamics, if enough time is allowed to elapse. Moreover, the experiments are not exactly reproducible, and the signal waveforms are quite complex. Consequently, determining accurately which sawteeth have been triggered by a previous notch in the RF power is not a simple statistical problem. Indeed, the incidence of random coincidences, i.e. sawteeth crashes that would have occurred anyway irrespective of the modulation, is difficult to quantify. Regarding the physics, the delay between the modulation and the occurrence of the sawteeth is a fundamental quantity to understand the details of the actual mechanism at play in this pacing scheme. This is another quantity not easy to determine with good accuracy.

The previous brief discussion should be enough to substantiate the statement that, in order to properly quantify the causal relationships in JET sawteeth pacing experiments with ICRH modulation, quite robust statistical techniques have to be deployed. Indeed, simple averages are indicators too vulnerable to outliers. Moreover, for experiments of this complexity and signals such as those of Fig. 13, simple Pearson correlation coefficients cannot be considered a proof of causation. The approach of Granger causality, which is based on predictability instead of correlation to determine causation, is certainly more appropriate. Unfortunately, the experiments on sawteeth pacing do not satisfy various assumptions of GC, such as linearity or separability of the causes. Therefore, the traditional GC techniques has proved to be quite ineffective. Recently more recent and different approaches have been tested, namely Recurrence Plots (RPs) [58], Convergent Cross Mapping (CCM) [59] and Transfer Entropy (TE) [60]. They have provided more than satisfactory results, but each one has its strengths and weaknesses [61,62]. All of them are quite delicate to implement and interpret and consequently a new approach based on autoencoders is a good and welcome complement.

In this case the analysis has been performed with two ensembles of 10 autoencoders each. Both are trained to predict the evolution of the core temperature: one using only the past of the temperature and the other with also the past the ICRH power as input. The autoencoders are trained to estimate the next time window of the $T_e$ signal. The time series of the ICRH power has been shifted forward in time to determine the delay of its effect on the sawteeth crash. The dataset has been divided in training set (80 % of the dataset) and test set (20 %). More specifically, the training set has been taken at the time windows from 8.5 s to 10.6 s and from 11.5 s to 13 s, while the test set is based on the time windows from 10.6 s to 11.5 s. The plots of Fig. 14 report the results of a scan in the backward window interval from 5 to 500 ms and for different sizes of the code.

The quality of the fits reported in Fig. 14 indicates clearly that the

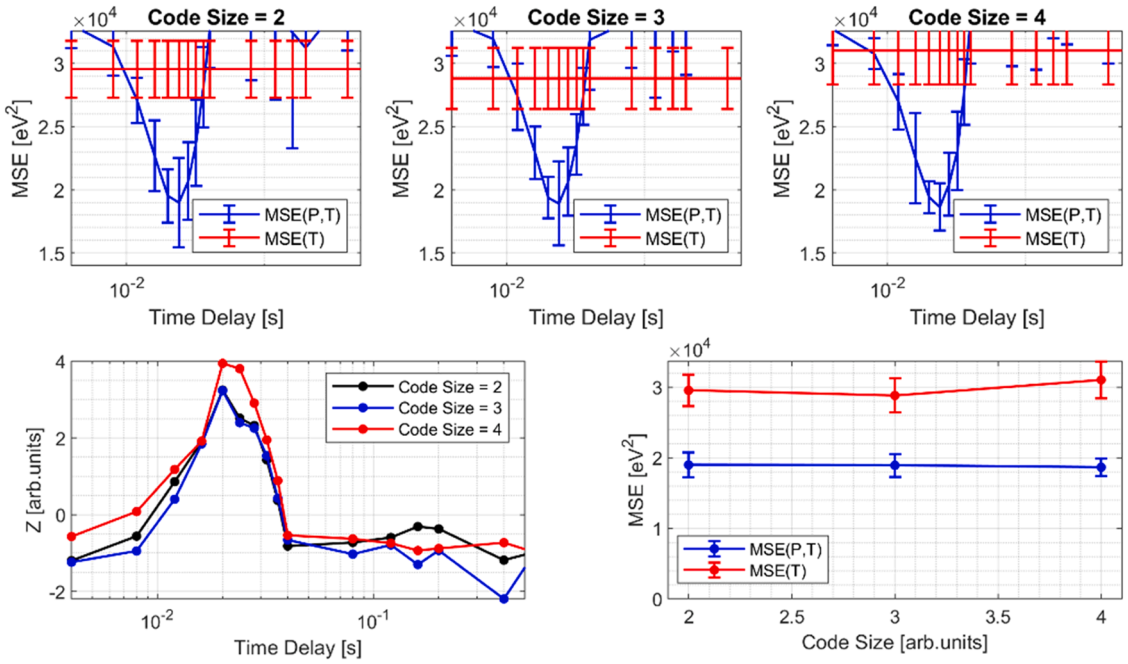


**Fig. 14.** Results obtained with two ensembles of 10 autoencoders each. The scan, the x axis, is derived varying the size of the backward time window. The comparison of the code size MSEs has been calculated for a delay of the ICRH equal to 20 ms.

modulation of the ICRH is an important input to the networks. Moreover, the MSE of the ensemble provided with the ICRH power as input shows a clear minimum for a delay of about $20 \pm 4$ ms between the ICRH notch and the crash of the next sawtooth. This result is perfectly in line with the physical interpretation of the experiments and the indications of the previously utilised tools, as reported in [54,63]. Indeed 20 ms is considered a good approximation of the fast ions slowing down time in these experiments. With regard to the code size, no significant difference has been observed for the interval investigated (see last plot of Fig. 14) and therefore from now on only the results of obtained with code size 3 are provided.

Fig. 15 shows the target temperature (indicated in black) against the autoencoder prediction, using only the past of the temperature signal (light blue) and also the past of the ICRH measurement (orange). In the top plot, the entire time window is shown, and it can be observed that blue predictions sometimes differ from the target. In the middle, the left and right plots report the predictions against the target for a sawteeth used to train the model (training set) and one sawteeth never previously seen by the autoencoder (test set). These two plots clearly show that the ensemble without ICRH is not able to correctly predict the sawteeth crashes in either case. Moreover, autoencoders with ICRH as additional input show smaller standard deviations, suggesting that the autoencoders of the ensemble have very similar predictions. The bottom plot shows the histogram of the logarithmic error between measurements and predictions.

In addition to complementing the techniques already deployed to analyse this data, the developed autoencoder technology allows investigating the entire spatiotemporal evolution of the experiments. Such a task can be performed by training the autoencoders to predict not only the core electron temperature but also the entire temperature profile. To this end two ensembles of 10 autoencoders each are trained to predict the evolution of the Electron Cyclotron Emission (ECE) diagnostic measurements. For each ECE channel, and therefore for each radial position, one ensemble is provided with only the past of the temperature as input, while the other is inputted also with the past the ICRH power. The autoencoders are trained to estimate the next time window of the $T_e$ signal. For each radial position, the time series of the ICRH power has been shifted forward in time to determine the delay of its effect on the sawteeth crash. The potential of the proposed methodology to investigate also the radial localisation of the sawteeth pacing scheme is shown graphically in Fig. 16, where the spatiotemporal evolution of the MSE for the fits of the temperature profile, with and without the ICRH power as input, is reported. It appears very clearly that the information about the driver reduces the MSE of the residuals in the time window about 20 ms before the crash, in agreement with the previous results and the literature. In terms of spatial distribution, the improvement of the fit is detected around a plasma radius of 3 m, which corresponds to the plasma core; this is in line with expectations since sawteeth are core instabilities that mainly affect the centre of the plasma column.

## Summary, conclusions and future developments

Autoencoders are a type of artificial neural network capable of learning fundamental features (codings) from unlabelled data through unsupervised learning. They achieve this by compressing the input space into a lower-dimensional representation (the code) and then reconstructing it using a decoder. This approach enables various efficient applications. By reducing dimensionality, autoencoders extract only the most relevant features, facilitating data analysis, clustering, and classification. Additionally, since the reconstructions retain only deterministic features, autoencoders can effectively denoise data. Other key applications include anomaly detection and density estimation.

In this work, autoencoder-based neural networks have been developed and tailored to address challenges in the analysis and modelling of

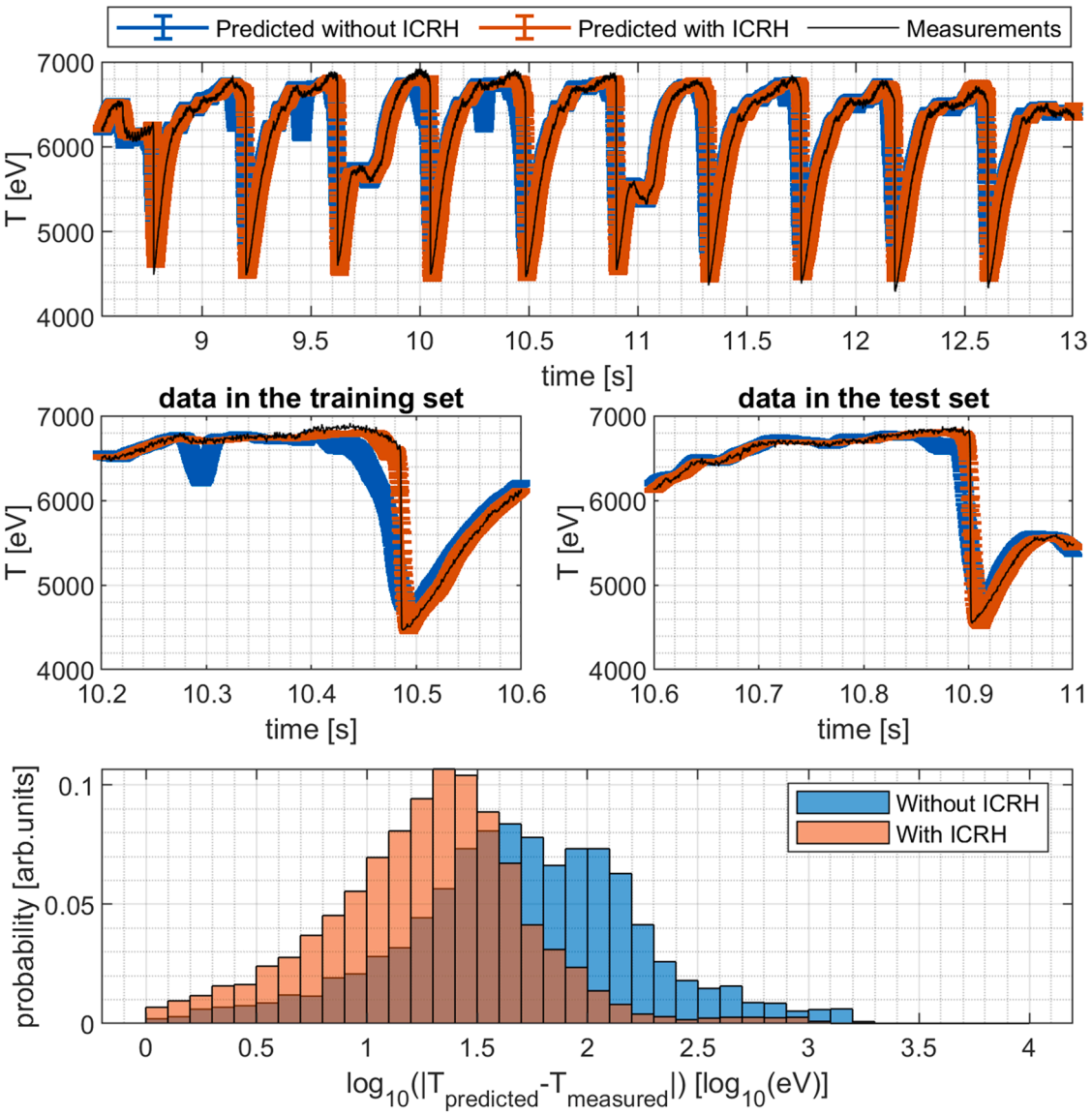


**Fig. 15.** Measured temperature against autoencoder predictions using as input only the temperature (blue) and also the ICRH (orange). The top plot shows the entire time-window analysed, while the second row shows two sawteeth crashes, the first one used in the training set and the second one in the test set. The last row shows the error distribution (in logarithm scale) between predicted and measured temperature, clearly showing that ICRH plays a relevant role in sawteeth crashes prediction. All the results have been obtained using the autoencoder with ICRH time delay equal to 20 ms.

time-dependent variables, with particular attention to the tasks of reconstructing hidden dynamics, modelling equations, and detecting causality.

The reconstruction of hidden variables and equation modelling have been achieved by incorporating a physics-informed (or model-informed) framework into the autoencoder architecture and training process (PIC-AE). This approach allows steering the learned representations towards a predefined physical or mathematical model. If the model parameters are unknown, they can be treated as learnable quantities, enabling model discovery. The PIC-AE has been tested numerically using the Lotka-Volterra system as a case study, demonstrating its ability to reconstruct actual (or equivalent) dynamics from indirect measurements. When applied to edge-localized modes in nuclear fusion, the PIC-AE has been unable to reconcile the experimental signals with the dynamics of a simple Lotka-Volterra system, suggesting the need to explore alternative models. It is worth mentioning that in the present work the analyses are limited to ordinary differential equations. However, the method could be extended to PDE systems by adopting higher-dimensional code representations, ranging from 0D for ODEs to 1D, 2D, or 3D arrays for PDEs in one, two, or three spatial dimensions, respectively; such an upgrade would enable the reconstruction of high-dimensional dynamical systems.

For causality detection, the methodology is based on Granger causality and an evolution of time-delay neural networks. Traditional TDNN-based approaches require large datasets and extensive parametric analysis to ensure reliability. To address these limitations, we have

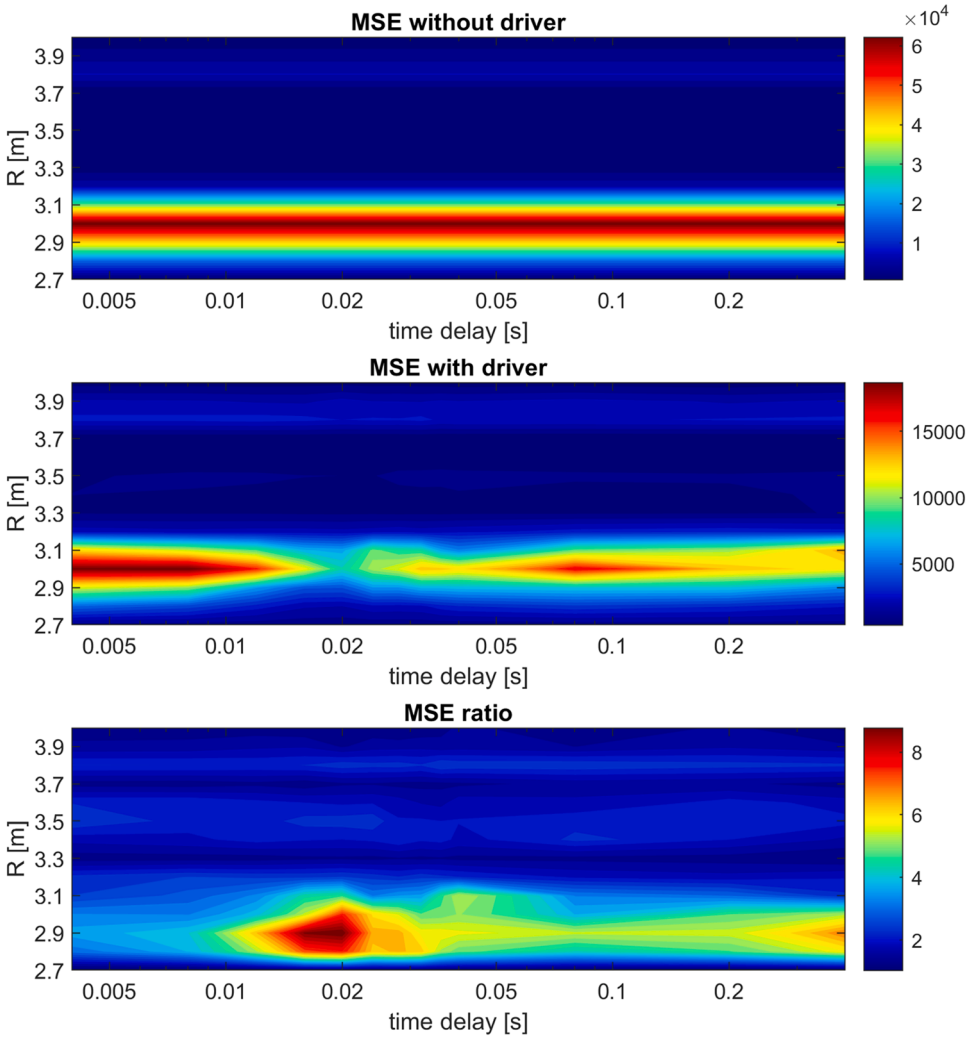


**Fig. 16.** The spatiotemporal evolution of the MSE for the fits with and without the ICRH power as input. It appears very clearly that the information about the driver reduces the MSE of the residuals in the time window about 20 ms before the crash and in the plasma core (radius around 3 m).

developed a novel autoencoder-based technique that accurately detects causal relationships without the need for extensive parametric tuning, while also providing a probabilistic measure of the coupling strength. This approach has been applied to nuclear fusion plasmas, specifically to the investigation of the causal relationship between sawtooth crashes (observed via electron temperature) and ion cyclotron resonance heating, explored as a possible control scheme for the is type of instability. The results are consistent with previous studies using different methodologies, validating both the autoencoder-based causality detection method and the findings related to sawtooth pacing. The potential of the developed tools is testified by a data driven spatio-temporal analysis of the sawteeth pacing mechanism, which to our knowledge has never been reported in the literature before.

Although developed for nuclear fusion applications, the methodology introduced in this work is highly generalizable and can be applied across various scientific and technological fields, in which time series analysis, understanding and modelling is important.

## CRediT authorship contribution statement

**R. Rossi:** Writing – original draft, Methodology, Investigation, Data curation. **A. Murari:** Writing – original draft, Methodology, Investigation, Data curation. **T. Craciunescu:** Writing – original draft, Methodology, Investigation, Data curation. **N. Rutigliano:** Writing – original draft, Methodology, Investigation, Data curation. **I. Wyss:** Writing – original draft, Methodology, Investigation, Data curation. **J. Vega:** Writing – original draft, Methodology, Investigation, Data curation. **P. Gaudio:** Writing – original draft, Methodology, Investigation, Data curation. **M. Gelfusa:** Writing – original draft, Methodology, Investigation, Data curation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgment**

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

## Appendix A. PIC-AE – Parametric analyses and additional information

This appendix shows the results related to some PIC-AE parametric analyses.

First, it is important to highlight again that the PIC-AE is not a surrogate model but a methodology aiming at analysing a specific time series and reconstructing its hidden dynamics. So, the model aim is specificity and not generalisation. Since it uses autoencoders and the latent space is constrained with physical equations, in principle the model cannot overfit the data. However, two distinct situations may occur:

- The weight of the physics is too large (see Fig. A.1, yellow case), and the model converges to a trivial solution. In this case, the error in the reconstruction of the data becomes unusually large (the goodness of fit tends to zero). This circumstance can be easily detected by inspection of the reconstruction loss.
- The weight of the physics is too small (see Fig. A.1, blue case), so the code will not reflect the physical equations. In this case, it may happen that the model tries to overfit part of the noise (even if most of it is removed by the autoencoder compression-decompression). In such a situation, the physics loss becomes large, and other runs, at higher physics weight, should be performed.

In order to avoid too many parametric cases, the adaptive weighting scheme adopted in the paper can help in finding directly a good solution. Fig. A.1 shows also the results achieved with the proposed adaptive weighting algorithm, showing that it converges on system coefficients with a minimum error compared to the target values.

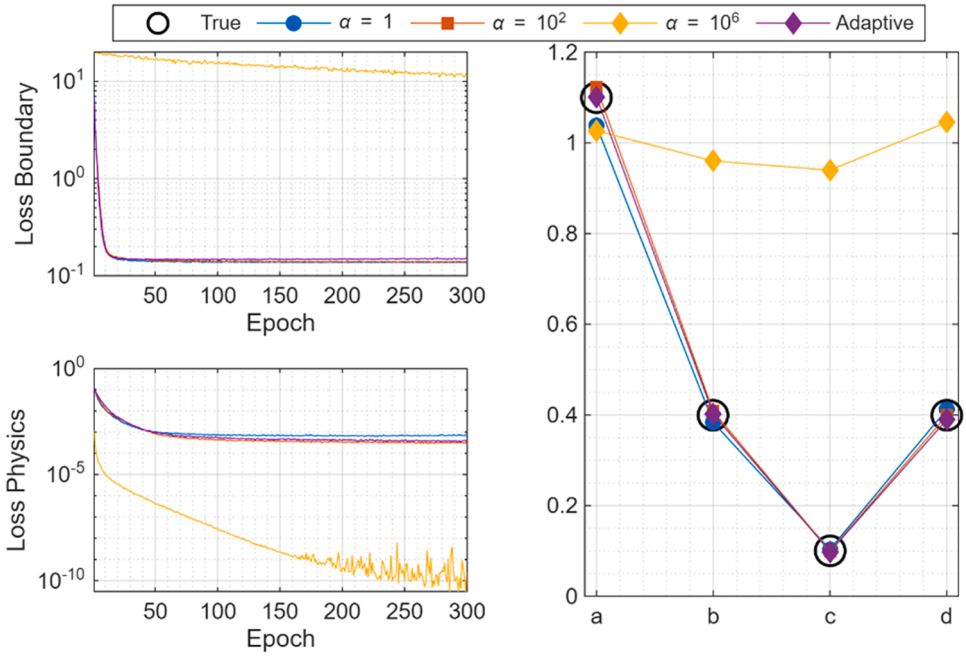


**Fig. A.1.** Evolution of loss boundary or measurements loss (top-left) and loss physics (bottom-left), and target and reconstructed system parameters (a, b, c and d, right) for the Lotka-Volterra case as a function of the weighting parameter α.

Regarding the architecture selection, our methodology is based on deep neural networks, since we need to ensure that both the encoder and the decoder are universal approximation functions for the family of models under investigation. To guarantee this, the procedure typically begins with a standard deep architecture and then the number of neurons in each layer is increased to verify that the autoencoder has sufficient depth. As this enlargement produces no noticeable gains in performance, we conclude that the corresponding design provides adequate representational capacity. The computational demand for the Lotka-Volterra case is 3 s per epoch (on a laptop with a NVIDIA GeForce RTX 4060 Laptop GPU) when 100 iterations per epoch are performed.

Fig. A.2 (left) reports the total loss (Eq. (12)) as a function of the learning rate. It can be shown that the loss decreases as a function of epochs except for a learning rate equal to $10^{-2}$, that is too high and does not allow for converge. Fig. A.2 (right) shows the target (black circles) against the reconstructed Lotka-Volterra parameters, confirming that for both good learning rates ($10^{-3}$ and $10^{-4}$) the reconstructed parameters converge to the target ones.

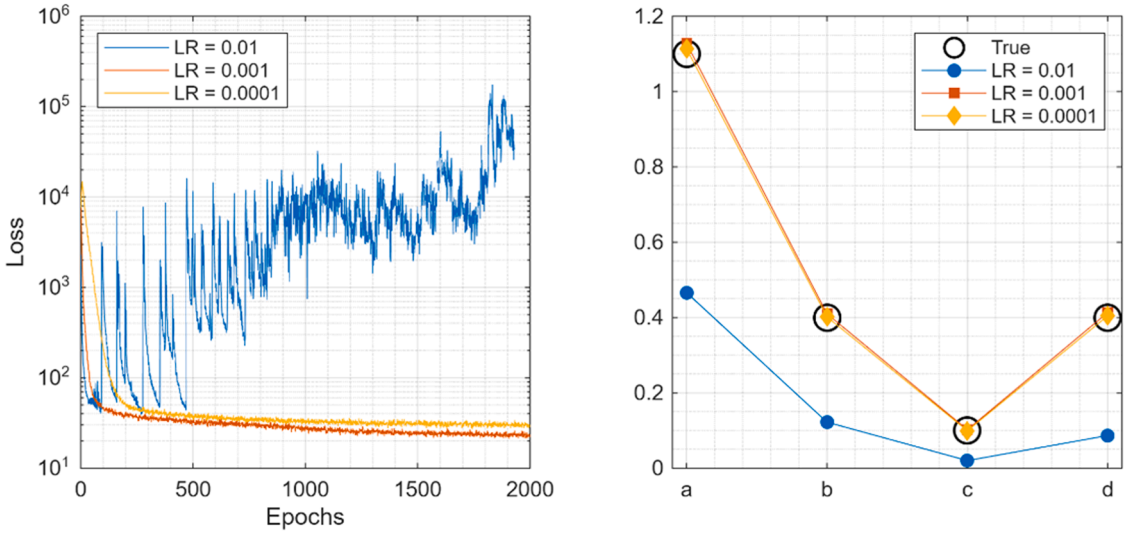


**Fig. A.2.** Total loss for the Lotka-Volterra system dynamic reconstruction using the PIC-AE (left) and target vs found parameters (right) as a function of learning rate.

Another analysis concerns the amount of data. In general, in chaotic systems, it is important that the data represents many orbits to ensure that the model learns the general dynamics and not only a limited part of it. We have performed a parametric analysis as a function of input data, 2000, 5000 and 10,000 points, which correspond to 1, 2.5 and 5 orbits. The results show an improvement in performances as a function of the orbit numbers, from a relative error around 33 % for 1 orbit to 20 % and 1.2 % for 2.5 and 5 orbits respectively.

The choice of the window (or buffer) size plays a crucial role, as it represents a trade-off between including enough points to mitigate the effect of noise and maintaining a sufficiently local representation of the dynamics. If the window is too small, noise dominates and the estimates become inaccurate; if it is too large, the PIC-AE is forced to approximate a non-local portion of the trajectory, typically resulting in larger errors. The optimal window size is inherently problem-dependent, as it is influenced by factors such as the characteristic timescales of the system, the temporal resolution of the measurements, and the noise level. Therefore, the selection should rely on physical insight, for example choosing the largest window that still remains within a locally coherent region of the attractor or time pattern, or through a dedicated parametric study. In our Lotka–Volterra example, we performed such a parametric analysis by varying the window length and evaluating the standard deviation of the inferred parameters, as shown in Fig. A.3. The results reveal a range of window sizes (approximately from 10 to 50 time steps) for which the outputs remain stable. In contrast, both smaller and larger windows exhibit significant variability, indicating a strong sensitivity to the window length outside this range.

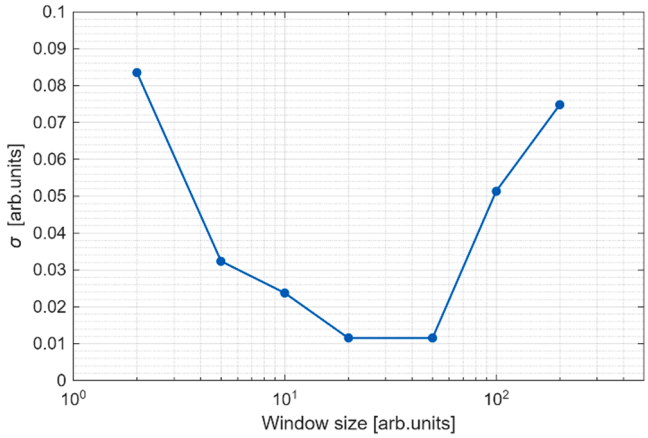


**Fig. A.3.** Output stability estimated as the standard deviation (σ) of the model parameters as a function of the window size.

The same kind of analysis can be applied to the minibatch size. In this case, a minibatch that is too small (e.g., $< 50$) can lead to unstable loss, while a very large minibatch increases computational demand and may cause overfitting or loss stagnation. Also in this case, a parametric analysis is suggested on any new problem, even if in all our cases we found that a minibatch equal to 1000 provides stable and accurate results.

Comparison with SINDy and DMD

The methods SINDy, DMD, Koopman, and PIC-AE all aim to reconstruct dynamical behaviour from observed data. However, their objectives and operating assumptions are fundamentally different, and therefore PIC-AE is not intended to replace these techniques but rather to complement them in scenarios, in which the others cannot operate.

In DMD and Koopman-based models, the goal is to obtain a linear representation of the evolution in an appropriate space of observables. These approaches provide an equivalent linear dynamical description but do not recover the underlying nonlinear physical model, unless the correct nonlinear observables are explicitly supplied.

Both SINDy and PIC-AE can, in principle, recover the actual governing equations, but only when suitable prior information is available. The key difference is that SINDy operates directly on the true system variables, assuming they are measured, whereas PIC-AE operates on proxy signals,

explicitly allowing for situations, in which the true system quantities are unknown or unmeasurable.

Consequently, a direct comparison between PIC-AE and DMD or Koopman models is not meaningful, as they produce fundamentally different outputs. A comparison between PIC-AE and SINDy is possible but must acknowledge that they operate under different information and measurement conditions. A comparison of the four methods properties is given in Table A.1.

**Table A.1**
Comparison between DMD, Koopman, SINDy and PIC-AE methods in terms of inputs and outputs.

| | DMD | Koopman | SINDy | PIC-AE |
|---|---|---|---|---|
| Requires direct measurement of true system variables? | Yes, if you want to model the observable and not proxy | Yes, if you want to model the observable and not proxy | Yes, if you want to model the observable and not proxy | No, the AE reconstruct the observables from proxies |
| Requires prior physical knowledge? | No | No | Yes, if you want to find the actual physical model. | Yes |
| What the method returns (output) | Dynamic modes, eigenvalues, linear reconstruction of the trajectory (not the physical model) | Finite-dimensional approximation of the Koopman operator; linear evolution in observable space; predictive but not the governing equations | Explicit sparse differential equations; estimated physical parameters; interpretable model | Latent physical state, inferred parameters, and imposed/learned nonlinear dynamics; reconstruction of hidden states |

To provide a quantitative assessment, we consider the Lotka–Volterra system presented in Section 2.2. In the SINDy case, we feed the hidden variables $(x, y)$ rather than the proxies $(p, q)$. A parametric noise analysis is carried out by adding noise proportional to the signal amplitude. Because the basic SINDy implementation used here is highly sensitive to noise, primarily because it must differentiate noisy signals, we place SINDy in a favourable condition, computing derivatives from ideal, noise-free trajectories. This setup makes SINDy's performance appear better than in realistic applications. This has been done since more advanced SINDy variants exist that are specifically designed to handle noise, but they are not considered in this comparison.

For PIC-AE, we report both the unconstrained case discussed in the main text and a parametric test in which the latent code is constrained using ten known points of $(x, y)$. Fig. A.4 shows the comparison. PIC-AE remains remarkably robust to noise, whereas SINDy deteriorates significantly even at low noise levels. This difference arises from two main factors:

1. SINDy is a purely data-driven method, while PIC-AE incorporates the known (or hypothesized) physics directly into the latent dynamics. Physics-informed constraints are known to reduce the influence of noise and outliers, at the cost of requiring prior physical knowledge, an inherent limitation of PIC-AE but also its main strength.
2. PIC-AE implements differentiation through automatic differentiation on the network-generated, smooth latent trajectories, whereas SINDy must differentiate the noisy measured signals directly.

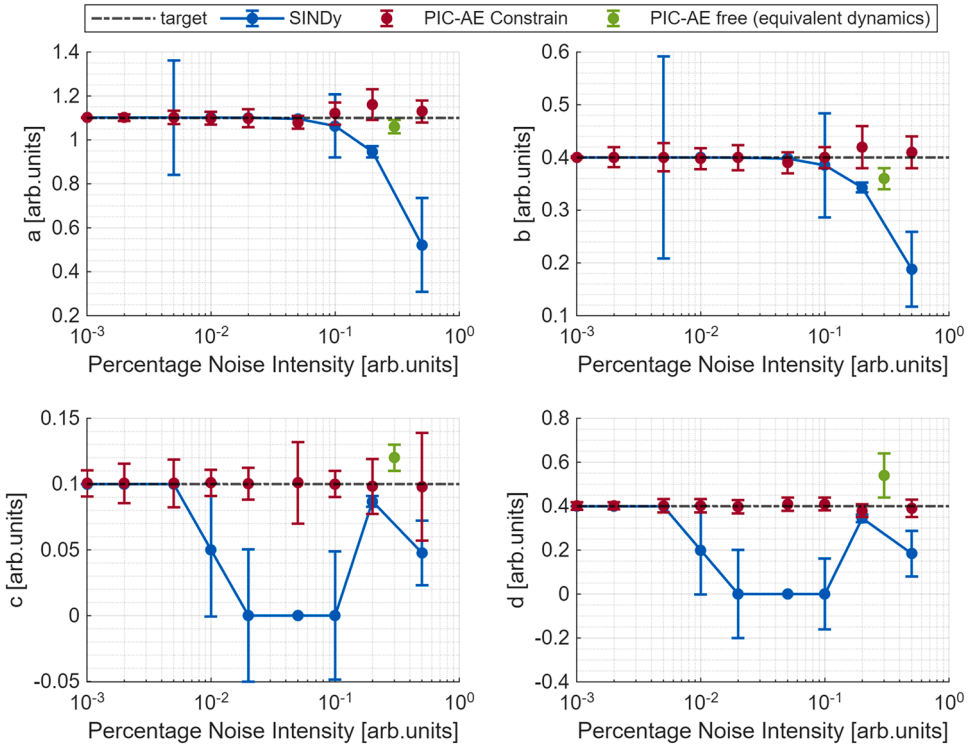


**Fig. A.4.** Comparison between SINDy and PIC-AE applied to the Lotka-Volterra system as a function of noise intensity.

## Appendix B. Causality autoencoders: synthetic tests and parametric analyses

Causality autoencoders are based on deep neural networks, since the methodology is based on the fact that the encoder and decoder are universal approximation functions. Following the approach adopted in PIC-AE autoencoders and related studies, we proceed by starting with a relatively shallow baseline autoencoder and progressively increase its complexity. Once additional layers and neurons fail to yield measurable performance gains, we conclude that the initial architecture provides sufficient depth for the specific problem. A decay learning rate scheme is used also in this case, and all the results have been obtained with a starting learning rate equal to 0.001 and a decay rate equal to 0.0001.

To ensure high reliability, instead of training just one autoencoder per case, the ensemble autoencoder methodology has been deployed. This of course has a drawback, which is the increase of computational time. If one autoencoder typically needs T seconds, the ensemble will need 2 M T seconds, where M is the number of couple of autoencoders in the ensemble and the factor of 2 takes into account the need to train one autoencoder of the couple with and one without the driver. The training time for one ensemble is very variable and depends on the complexity and dimensionality of the problem. For the synthetic cases below, each point statistically requires 1 min (M is equal to 3, the validation checks equal to 10, iterations per epoch equal to 100, and the minibatch equal to 1000). In the experimental case reported in the paper, the parametric scan as a function of time delay requires 20 min (in this case M is equal to 10), while the parametric analysis as a function of both time delay and position takes around 5 h (same hyperparameters).

A systematic series of tests has been performed also for the autoencoders designed to test the causal relationships between time series. Some of the most informative cases are reported in the following. They are typical examples considered as benchmarks for this type of applications, as reported in [44].

**Autoregressive (AR) model:**

The following equations describe an autoregressive model:

$$x(t) = 0.5x(t-1) + 0.2y(t-1) + \epsilon_x(t)$$

$$y(t) = Cx(t-1) + 0.7y(t-1) + \epsilon_y(t) \tag{B1}$$

In this case, the relevant point to notice is that the influence of $x$ on $y$ depends on the coupling parameter $C$. The result obtained from a systematic scan of $C$ is shown in the plots of Fig. B1. There are various important aspects to observe in this figure. First, when the coupling increases the quality of the fit obtained by the AE without the input x decreases systematically. This trend is not observed for the AE which uses also the information of $x$. The indicator $Z$, which summarises the statistical prediction performance difference between the two AE, increases proportionally to $C$, indicating that the autoencoder with the input $x$ has identified the proper dependence between the two time series. Finally, in case of zero coupling the $Z$ test properly detects no statically relevant causation between the two time series: this fact indicates that the autoencoders are not vulnerable to spurious influences.

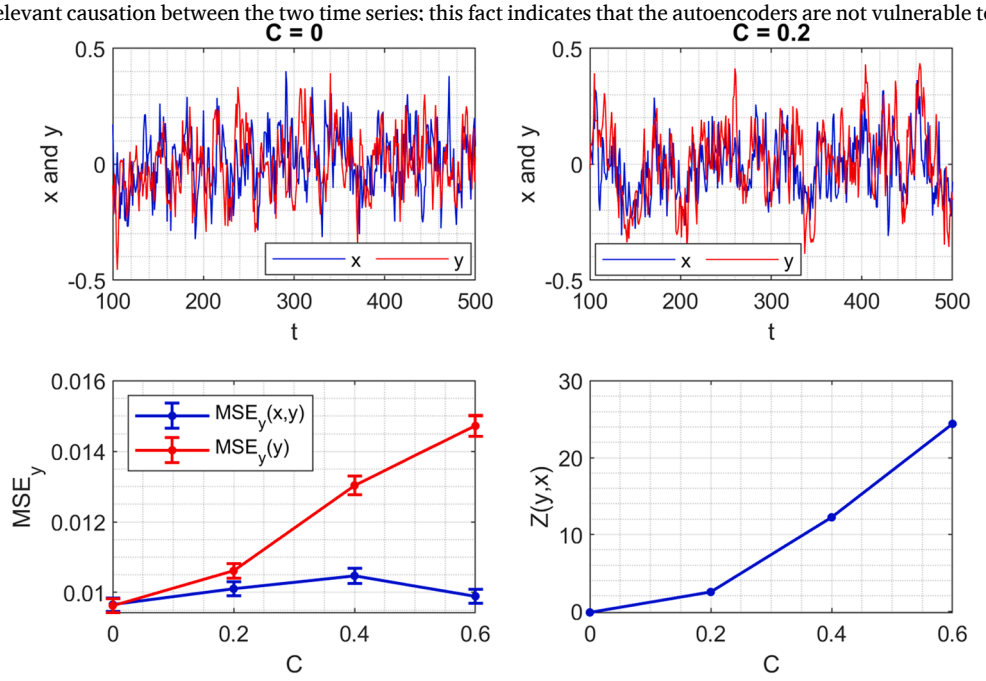


**Fig. B1.** Top row. Autoregressive system in case of $C = 0$ (x does not cause y) and $C = 0.2$ (x causes y) (top-left and top-right respectively). Bottom row. Prediction MSE of the two autoencoders (bottom-left) and indicator Z (bottom-right) as a function of C.

A parametric analysis as a function of dataset size ($N = 1000$, 2000, 5000 and 10,000) has been performed for the AR system and the values of Z, and statistical results are shown in Table B1 (results have been obtained using an ensemble size equal to three). The results show that the methodology returns good causality detection in case of $C = 0$ and strong coupling factors ($C = 0.4$ and 0.6), while confusion occurs for weak coupling factors in the case of smaller dataset size. The reason lays in the training process. If the dataset is too small, the autoencoder is not able to generalise correctly and may return unreliable results.

**Table B1**
Z values for coupled AR system as a function of coupling factor C and dataset size N.

| N | $C = 0$ | $C = 0.2$ | $C = 0.4$ | $C = 0.6$ |
|---|---|---|---|---|
| 1000 | −1.67 | 0.37 | 5.74 | 83.05 |
| 2000 | −0.46 | −0.55 | 7.85 | 9.57 |
| 5000 | −0.54 | 1.52 | 13.66 | 22.31 |
| 10,000 | 0.07 | 2.76 | 11.10 | 24.35 |

Test Henon-Henon system

The coupling of two Henon systems, both in the chaotic regime, is another typical benchmark to assess the potential of causality identification techniques well reported and discussed in the literature [44]. The detailed set of Eqs. (B2) has been analysed again as a function of the coupling parameter $C$.

$$x_1(t+1) = 1.4 - x_1^2(t) + 0.3x_2(t)$$

$$x_2(t+1) = x_1(t)$$

$$y_1(t+1) = 1.4 - \left[Cx_1(t)y_1(t) + (1-C)y_1^2(t)\right] + 0.3y_2(t)$$

$$y_2(t+1) = y_1(t) \tag{B2}$$

The results obtained with the proposed autoencoder architecture are summarised in the plots of Fig. B2. In this case the interpretation of the results is a bit more involved. Indeed, at the beginning the prediction performances without x decreases drastically (MSE increases) with the increase of the coupling coefficient $C$. This is also observed by the indicator $Z$. Then, there is almost a saturation of Z, followed by its drop for a coupling coefficient equal to 0.72. This is due to the fact that, for values of C equal or higher than 0.7, the two systems are perfectly synchronised. Consequently, the information in x to predict y is already fully contained in y and therefore the model cannot detect any causality. This behaviour is observed by all causality methods and is discussed in detail in [44].

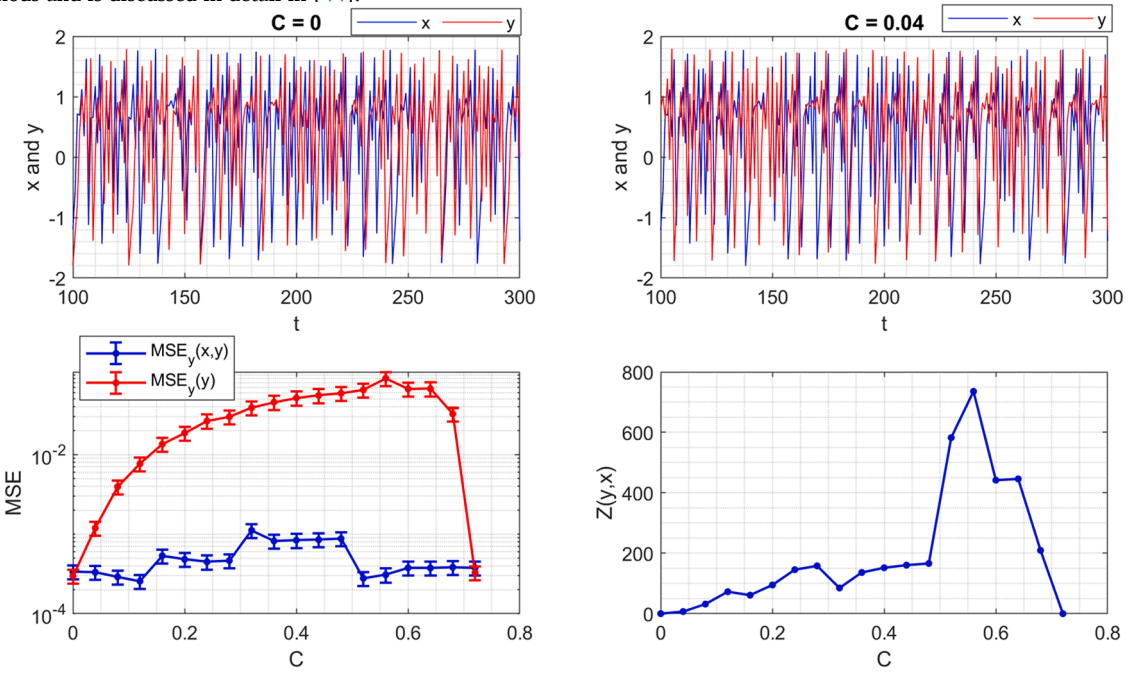


**Fig. B2.** Top row. Henon-Henon system in case of $C = 0$ (x does not cause y) and $C = 0.04$ (x causes y) (top-left and top-right respectively). Bottom row. Prediction MSE of the two autoencoders (bottom-left) and indicator Z (bottom-right) as a function of C.

## Data availability

Codes for analysis and synthetic data generation are available at the following link: https://github.com/QEP-Repository/PICAE_CausAE">https://github.com/QEP-Repository/PICAE_CausAE Experimental data are available upon request to the authors and after the permission from EUROfusion consortium.

## References

[1] B.P. Bezruchko, D.A. Smirnov, Extracting Knowledge From Time Series, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, https://doi.org/10.1007/978-3-642-12601-7.
[2] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed., Wiley, 2015.
[3] C.C. Holt, Forecasting seasonals and trends by exponentially weighted moving averages, Int. J. Forecast. 20 (1) (Jan. 2004) 5–10, https://doi.org/10.1016/j.ijforecast.2003.09.015.

[4] D. Asteriou, S.G. Hall, Vector autoregressive (VAR) models and causality tests. Applied Econometrics, Macmillan Education UK, London, 2016, pp. 333–346, https://doi.org/10.1057/978-1-137-41547-9_15.

[5] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (Nov. 1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.

[6] J. Durbin, S.J. Koopman, Time Series Analysis By State Space Methods, Oxford University Press, 2012, https://doi.org/10.1093/acprof:oso/9780199641178.001.0001.

[7] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2005, https://doi.org/10.7551/mitpress/3206.001.0001.

[8] K. Zhao, et al., Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: a bayesian ensemble algorithm, Remote Sens. Environ. 232 (Oct. 2019) 111181, https://doi.org/10.1016/j.rse.2019.04.034.

[9] S.J. Taylor and B. Letham, "Forecasting at scale," Sep. 27, 2017. doi: 10.7287/peerj.preprints.3190v2.

[10] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (15) (Apr. 2016) 3932–3937, https://doi.org/10.1073/pnas.1517384113.

[11] S.L. Brunton, J.N. Kutz, Data-Driven Science and Engineering, Cambridge University Press, 2022, https://doi.org/10.1017/9781009089517.

[12] J.H. Tu, C.W. Rowley, D.M. Luchtenburg, S.L. Brunton, J.Nathan Kutz, On dynamic mode decomposition: theory and applications, J. Computat. Dyn. 1 (2) (2014) 391–421, https://doi.org/10.3934/jcd.2014.1.391.

[13] P.J. SCHMID, Dynamic mode decomposition of numerical and experimental data, J. Fluid. Mech. 656 (Aug. 2010) 5–28, https://doi.org/10.1017/S0022112010001217.

[14] A. Mauroy, I. Mezić, Y. Susuki (Eds.), The Koopman Operator in Systems and Control, The Koopman Operator in Systems and Control, 484, Springer International Publishing, Cham, 2020, https://doi.org/10.1007/978-3-030-35713-9.

[15] P. Bevanda, S. Sosnowski, S. Hirche, Koopman operator dynamical models: learning, analysis and control, Annu. Rev. Control 52 (2021) 197–212, https://doi.org/10.1016/j.arcontrol.2021.09.002.

[16] D. Bank, N. Koenigstein, R. Giryes, Autoencoders. Machine Learning for Data Science Handbook, Springer International Publishing, Cham, 2023, pp. 353–374, https://doi.org/10.1007/978-3-031-24628-9_16.

[17] F.F. Chen, Introduction to Plasma Physics and Controlled Fusion, Springer International Publishing, 2016. Third Edition.

[18] J. Wesson, Tokamaks, Oxford University Press, 2011.

[19] T.C. Hender, et al., Chapter 3: MHD stability, operational limits and disruptions, Nucl. Fus. 47 (6) (Jun. 2007) S128–S202, https://doi.org/10.1088/0029-5515/47/6/S03.

[20] H. Zohm (Ed.), Magnetohydrodynamic Stability of Tokamaks, Wiley, 2014, https://doi.org/10.1002/9783527677375.

[21] D.N. Hill, A review of ELMs in divertor tokamaks, J. Nucl. Mater. (Feb. 1997) 241–243, https://doi.org/10.1016/S0022-3115(97)80039-6.

[22] F. Fainstein, G.B. Mindlin, P. Groisman, Reconstructing attractors with autoencoders, Chaos: Interdiscip. J. Nonlinear Sci. 35 (1) (Jan. 2025), https://doi.org/10.1063/5.0232584.

[23] C. Lai, P. Baraldi, E. Zio, Physics-informed deep autoencoder for fault detection in new-design systems, Mech. Syst. Signal. Process. 215 (Jun. 2024) 111420, https://doi.org/10.1016/j.ymssp.2024.111420.

[24] W. Zhong, H. Meidani, PI-VAE: physics-informed variational auto-encoder for stochastic differential equations, Comput. Methods Appl. Mech. Eng. 403 (Jan. 2023) 115664, https://doi.org/10.1016/j.cma.2022.115664.

[25] N. Rutigliano, et al., Physics-informed neural networks for the modelling of interferometer-polarimetry in tokamak multi-diagnostic equilibrium reconstructions, Plasma Phys. Control Fusion. 67 (6) (Jun. 2025) 065029, https://doi.org/10.1088/1361-6587/addde6.

[26] R. Rossi, M. Gelfusa, A. Murari, On the potential of physics-informed neural networks to solve inverse problems in tokamaks, Nucl. Fus. 63 (12) (Dec. 2023) 126059, https://doi.org/10.1088/1741-4326/ad067c.

[27] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: 3rd International Conference for Learning Representations, San Diego, 2015.

[28] A. Pavone, A. Merlo, S. Kwak, J. Svensson, Machine learning and bayesian inference in nuclear fusion research: an overview, Plasma Phys. Control Fus. 65 (5) (May 2023) 053001, https://doi.org/10.1088/1361-6587/acc60f.

[29] H. Zohm, Magnetohydrodynamic Stability of Tokamaks, Wiley-VCH, 2014.

[30] Valentin Igochine, Active Control of Magneto-Hydrodynamic Instabilities in Hot Plasmas, 83, Berlin Heidelberg, 2015.

[31] H. Zohm, Edge localized modes (ELMs), Plasma Phys. Control Fus. 38 (2) (Feb. 1996), https://doi.org/10.1088/0741-3335/38/2/001.

[32] A.W. Leonard, Edge-localized-modes in tokamaks, Phys. Plasmas. 21 (9) (Sep. 2014), https://doi.org/10.1063/1.4894742.

[33] J.W. Connor, Edge-localized modes - physics and theory, Plasma Phys. Control Fus. 40 (5) (May 1998), https://doi.org/10.1088/0741-3335/40/5/002.

[34] A. Cathey, et al., Comparing spontaneous and pellet-triggered ELMs via non-linear extended MHD simulations, Plasma Phys. Control Fusion. 63 (7) (Jul. 2021), https://doi.org/10.1088/1361-6587/abf80b.

[35] D. Frigione, et al., Divertor load footprint of ELMs in pellet triggering and pacing experiments at JET, J. Nucl. Mater. 463 (Aug. 2015) 714–717, https://doi.org/10.1016/j.jnucmat.2015.01.048.

[36] J. Mailloux, et al., Overview of JET results for optimising ITER operation, Nucl. Fus. 62 (4) (Apr. 2022) 042026, https://doi.org/10.1088/1741-4326/ac47b4.

[37] C.F. Maggi, et al., Overview of T and D–T results in JET with ITER-like wall, Nucl. Fus. 64 (11) (Nov. 2024) 112012, https://doi.org/10.1088/1741-4326/ad3e16.

[38] D.M. Eagleman, A.O. Holcombe, Causality and the perception of time, Trends. Cogn. Sci. 6 (8) (Aug. 2002) 323–325, https://doi.org/10.1016/S1364-6613(02)01945-9.

[39] C.W.J. Granger, Some recent development in a concept of causality, J. Econom. 39 (1–2) (Sep. 1988) 199–211, https://doi.org/10.1016/0304-4076(88)90045-0.

[40] J. Losee, Theories of Causality, Routledge, 2017, https://doi.org/10.4324/9781315135533.

[41] C.W.J. Granger, Time series analysis, cointegration, and applications, Am. Econ. Rev. 94 (3) (May 2004) 421–425, https://doi.org/10.1257/0002828041464669.

[42] C.W.J. Granger, Investigating causal relations by econometric models and cross-spectral methods, Econometrica 37 (3) (Aug. 1969) 424, https://doi.org/10.2307/1912791.

[43] A. Shojaie, E.B. Fox, Granger causality: a review and recent advances, Annu Rev. Stat. Appl. 9 (1) (Mar. 2022) 289–319, https://doi.org/10.1146/annurev-statistics-040120-010930.

[44] A. Krakovská, J. Jakubík, M. Chvosteková, D. Coufal, N. Jajcay, M. Paluš, Comparison of six methods for the detection of causality in a bivariate time series, Phys. Rev. e 97 (4) (Apr. 2018) 042207, https://doi.org/10.1103/PhysRevE.97.042207.

[45] V. Peddinti, D. Povey, S. Khudanpur, A time delay neural network architecture for efficient modeling of long temporal contexts. Interspeech 2015, ISCA: ISCA, Sep. 2015, pp. 3214–3218, https://doi.org/10.21437/Interspeech.2015-647.

[46] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang, Phoneme recognition using time-delay neural networks, IEEe Trans. Acoust. 37 (3) (Mar. 1989) 328–339, https://doi.org/10.1109/29.21701.

[47] A. Murari, R. Rossi, M. Gelfusa, Combining neural computation and genetic programming for observational causality detection and causal modelling, Artif. Intell. Rev. 56 (7) (Jul. 2023) 6365–6401, https://doi.org/10.1007/s10462-022-10320-3.

[48] S. von Goeler, W. Stodiek, N. Sauthoff, Studies of Internal disruptions and <math display="inline"><mi>m</mi><mo>=</mo><mn>1</mn><mn/></math>oscillations in Tokamak discharges with soft—X-Ray tecniques, Phys. Rev. Lett. 33 (20) (Nov. 1974) 1201–1203, https://doi.org/10.1103/PhysRevLett.33.1201.

[49] I.T. Chapman, et al., Empirical scaling of sawtooth period for onset of neoclassical tearing modes, Nucl. Fus. 50 (10) (Oct. 2010) 102001, https://doi.org/10.1088/0029-5515/50/10/102001.

[50] J.P. Graves, et al., Sawtooth control in JET with ITER relevant low field side resonance ion cyclotron resonance heating and ITER-like wall, Plasma Phys. Control Fus. 57 (1) (Jan. 2015) 014033, https://doi.org/10.1088/0741-3335/57/1/014033.

[51] E. Lerche, et al., Optimization of ICRH for core impurity control in JET-ILW, Nucl. Fus. 56 (3) (Mar. 2016) 036022, https://doi.org/10.1088/0029-5515/56/3/036022.

[52] T.P. Goodman, F. Felici, O. Sauter, J.P. Graves, Sawtooth pacing by real-time auxiliary power control in a Tokamak plasma, Phys. Rev. Lett. 106 (24) (Jun. 2011) 245002, https://doi.org/10.1103/PhysRevLett.106.245002.

[53] S.E. Sharapov, et al., Experimental studies of instabilities and confinement of energetic particles on JET and MAST, Nucl. Fus. 45 (9) (Sep. 2005) 1168–1177, https://doi.org/10.1088/0029-5515/45/9/017.

[54] E. Lerche, et al., Sawtooth pacing with on-axis ICRH modulation in JET-ILW, Nucl. Fus. 57 (3) (Mar. 2017) 036027, https://doi.org/10.1088/1741-4326/aa53b6.

[55] E. Lerche, et al., Sawtooth control with modulated ICRH in JET-ILW H-mode plasmas, Nucl. Fus. 60 (12) (Dec. 2020) 126037, https://doi.org/10.1088/1741-4326/abb424.

[56] E. Lerche, et al., ICRH for core impurity mitigation in JET-ILW, in: AIP Conference Proceedings, 2015 030002, https://doi.org/10.1063/1.4936467.

[57] E. Lerche, et al., Sawtooth pacing with on-axis ICRH modulation in JET-ILW, Nucl. Fus. 57 (3) (Mar. 2017) 036027, https://doi.org/10.1088/1741-4326/aa53b6.

[58] N. Marwan, M. Carmenromano, M. Thiel, J. Kurths, Recurrence plots for the analysis of complex systems, Phys. Rep. 438 (5–6) (Jan. 2007) 237–329, https://doi.org/10.1016/j.physrep.2006.11.001.

[59] G. Sugihara, et al., Detecting causality in complex ecosystems, Science (1979) 338 (6106) (2012) 496–500, https://doi.org/10.1126/science.1227079. Oct.

[60] T. Schreiber, Measuring information transfer, Phys. Rev. Lett. 85 (2) (Jul. 2000) 461–464, https://doi.org/10.1103/PhysRevLett.85.461.

[61] A. Murari, et al., How to assess the efficiency of synchronization experiments in tokamaks, Nucl. Fus. 56 (7) (Jul. 2016) 076008, https://doi.org/10.1088/0029-5515/56/7/076008.

[62] A. Murari, T. Craciunescu, E. Peluso, E. Lerche, M. Gelfusa, On efficiency and interpretation of sawteeth pacing with on-axis ICRH modulation in JET, Nucl. Fus. 57 (12) (Dec. 2017) 126057, https://doi.org/10.1088/1741-4326/aa87e7.

[63] E. Peluso, et al., Conditional recurrence plots for the investigation of sawteeth pacing with RF modulation, Plasma Phys. Control Fus. 64 (8) (Aug. 2022) 084002, https://doi.org/10.1088/1361-6587/ac757c.