SN Computer Science

Classification of trajectories of dynamic systems using physically-informed neural networks

--Manuscript Draft--

Manuscript Number:	SNCS-D-24-08760
Full Title:	Classification of trajectories of dynamic systems using physically-informed neural networks
Article Type:	Original Research
Section/Category:	Machine Learning
Funding Information:	
Abstract:	The paper proposes a method for multivariate time series classification. It uses information about their physical nature. The paper proposes a classification model for multidimensional trajectories of physical systems. In comparison to models that use assumptions about statistical properties of trajectories only, but do not use prior information, the presented model includes information about the physical nature of the system. It makes the classification stable to random non-physical changes in trajectories. To account for these changes we train the model on augmented samples. We propose to classify not the trajectories themselves, but the dynamical systems given by the Lagrangians corresponding to these trajectories. The Lagrangian is reconstructed using physics-informed Lagrangian neural networks. We introduce a norm in the space of Lagrangians and use it in metric classification.
Corresponding Author:	Alexander Terentyev Forecsys Ltd RUSSIAN FEDERATION
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Forecsys Ltd
Corresponding Author's Secondary Institution:	
First Author:	Alexander Terentyev
First Author Secondary Information:	
Order of Authors:	Alexander Terentyev
	Vadim Strijov
Order of Authors Secondary Information:	
Author Comments:	
Suggested Reviewers:	

Classification of trajectories of dynamic systems using physically-informed neural networks

Alexander Terentyev[1*] and Vadim Strijov[1*]

[1*]Forecsys LLC, Moscow, Russia.

*Corresponding author(s). E-mail(s): sasha-terentev@outlook.com; strijov@forecsys.ru;

Abstract

The paper proposes a method for multivariate time series classification. It uses information about their physical nature. The paper proposes a classification model for multidimensional trajectories of physical systems. In comparison to models that use assumptions about statistical properties of trajectories only, but do not use prior information, the presented model includes information about the physical nature of the system. It makes the classification stable to random non-physical changes in trajectories. To account for these changes we train the model on augmented samples. We propose to classify not the trajectories themselves, but the dynamical systems given by the Lagrangians corresponding to these trajectories. The Lagrangian is reconstructed using physics-informed Lagrangian neural networks. We introduce a norm in the space of Lagrangians and use it in metric classification.

Keywords: *physical system; time series; Lagrangian; Lagrangian neural network; compactness hypothesis*

1 Introduction

The physics-informed neural networks (PINNs) combine the advantages of statistical and mathematical models. The first ones fit the measured data while the second ones take a physical description of the modeled system into account. The PINNs model conservation laws using data as the basis for predictions. Its architecture is analyzed in [1]. PINNs model principles of Hamiltonian [2], Lagrangian dynamics [3], including Hamiltonian mechanics with infinite-dimensional measurement space [4] from quantum physics. The Lagrangian networks (LNN) also adapt to model microsystems using graph networks [5]. The modifications of LNN model complex macro-systems such as fluids with the multi-particle approximation [6]. Also, PINNs allow dimension reduction using special operator [7]. PINNs and LNNs also reconstruct sound [8] taking into account wave equations.

The paper [9] introduces Hamiltonian neural networks to adapt a nonlinear variation of a parametrized system and determine the parameter of nonlinearity. This parameter is responsible for the chaotic nature of the system, acting as a measure of chaos and separating the chaotic system from the original system. The disadvantage of this classification is that one needs to know in advance the law by which these systems are separated. According to the [10], the Neural Ordinary Differential Equations (NODE) could be unstable due to changes in the initial data, and the error grows exponentially over time. The architecture of CH-NODE reduces these shortcomings. This research focuses on the classification problem for systems with Lagrangian dynamics.

The convolutional neural networks are used for multivariate classifications. It assumes that there is a statistical relationship between the closest points of the time series [11]. These methods perform well in many tasks where these relationships are predominant. For physical systems such methods are not suitable. We propose using information about the physical connections of systems. For the designated Lagrangian neural networks this knowledge is the law of conservation of energy.

This paper solves the problem of dynamic system trajectory classification. It proposes to use Lagrangian neural networks (LNN) for low-dimensional trajectories. Each trajectory refers to the Lagrangian of the system parameterized by the LNN. The issue of mapping the Lagrangian of the system along the trajectory is analyzed in [3].

The dynamical system's state and trajectory are constructed with multivariate time serties [11]. The deep learning methods, developed for the univariate time series, like AlexNet, ResNet, InceptionTime, TapNet [11, 12] need architecture updates for the data acquired from several sensors [13] including three-axis accelerometer and gyroscope [14] and updates for the knowledge about the physical description of the dynamic system. The state-of-the-art approach in this field uses randomized kernels, such as ROCKET. However, it doesn't outperform others on all datasets [11, 15].

The Lagrangian network reconstructs system trajectories where the conservation of energy and the Euler-Lagrange equations hold. It uses an independent set of coordinates to specify the state of the system. In the Lagrangian formalism, the dynamic system is defined by the Lagrangian function. The generalized coordinates in Lagrangian mechanics specify the state of the system. The minimum number of independent variables (generalized coordinates) necessary for a description of the state of the mechanical system is chosen. The Lagrangian of the system is a continuous differentiable and square-integrable function. Its Euler-Lagrange equation has a unique solution. The dynamic system is described by the Euler-Lagrange equation,

$$\frac{\partial L}{\mathbf{q}} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = Q_x = 0. \tag{1}$$

For its unique solution, the necessary and sufficient condition is

$$\det \left\| \frac{\partial^2 L}{\partial \dot{q}_j \partial \dot{q}_k} \right\|_{j,k=1}^{n} \neq 0. \tag{2}$$

The equivalent physical definition of the Lagrangian is the difference between the kinetic T and potential V energy,

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q}, \dot{\mathbf{q}}), \tag{3}$$

where $\mathbf{q}_t$ are the generalized coordinates of the system, $\dot{\mathbf{q}}_t$ are the generalized accelerations of the system, L is the Lagrangian of the system. For Lagrangians of different motions of a physical system, we assume the compactness hypothesis: the Lagrangians which correspond to different motions lie in different compact and separable subsets. We label these subsets as the classes in the space of trajectory models. We assume the parameters of these models are also separable for different classes. The computational experiment uses the PAMAP2 dataset [16]. The dataset contains trajectories of human movements during common daily activities like cooking, jogging, and shopping.

The dataset contains recordings from three sets of gyroscopes and accelerometers: attached to the wrist of the dominant arm, attached to the chest, and attached to the elbow of the predominant hand. The number of subjects is 9. The number of activity types(classes) is 24, each activity lasted 2–4 minutes [16]. The sampling frequency is 100 Hz. This dataset provides an opportunity to evaluate the performance of the algorithm on a complex dynamic system. We consider general coordinates for the system that is described by the coordinates of all sensors, which are used in the experiments, as one vector.

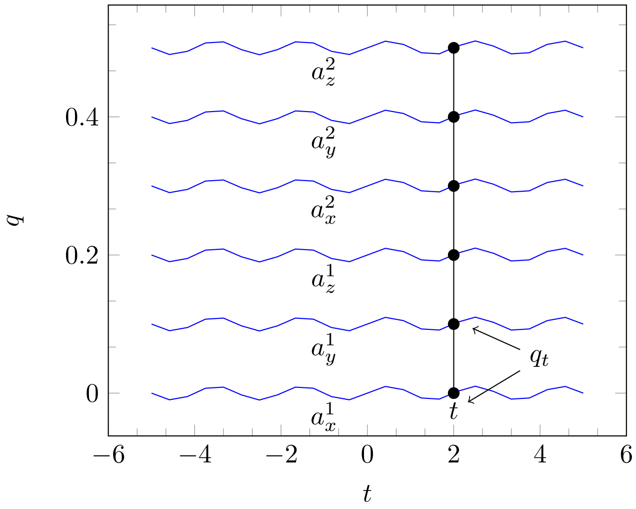

Figure 1 A point in the phase space as a sample of the dataset

2 Regression problem statement for Lagrangian networks

Assume that the Lagrangian L of the dynamical system corresponds to some specific trajectory. The Lagrangian defines a dynamical system for which the trajectory is constructed: substituting L into the Euler-Lagrange equation (1) yields a function relating the velocity and coordinates of the point to its acceleration. The solution of the ODE system is obtained by substituting L into (1), with initial conditions, the initial point of the trajectory, and the trajectory is fully reconstructed. We identify each trajectory by some function L from the space with norm L_2 and identify the trajectory and the dynamical system, given by the function L, which we further feed to the input of the classifier. The problem of recovery L we formulate as the regression problem.

Denote by X the trajectory of dimension r and length T

$$X = [\mathbf{X}_1, \dots, \mathbf{X}_T] \quad \text{where} \quad \mathbf{x}_i = (\mathbf{q}_i, \dot{\mathbf{q}}_i), \quad \text{and} \quad \dot{\mathbf{q}}_i \in \mathbb{R}^r \tag{4}$$

is the velocity at the i-th time-tick, and $\mathbf{q}_i \in \mathbb{R}^r$ is the coordinate at the i-th time-tick.

There given a sample set

$$D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N, \tag{5}$$

where X are trajectories of dimension r and length T, $\mathbf{y}_i = \ddot{\mathbf{q}}_i \in \mathbb{R}^r$ is the acceleration at the i-th time-tick. One has to find a function that predict acceleration for particular point, using previous data D

$$\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}|D), \quad \text{where} \quad \hat{Y} = \{\hat{\mathbf{y}}_i = \hat{\ddot{\mathbf{q}}}_i\}_{i=1}^N \tag{6}$$

is the predicted trajectory of the dynamic system. The function $\mathbf{f} = \mathbf{f}(\mathbf{x}|\mathbf{w})$ with parameters $\mathbf{w}$ is selected from the family of Lagrangian neural networks.

The loss function of the predicted acceleration $\hat{\mathbf{y}}_i$ given the points $\mathbf{X}_i = (\mathbf{q}_i, \dot{\mathbf{q}}_i)$ of the trajectory X is

$$\mathcal{L}(\mathbf{w}) = \mathcal{L}(\mathbf{w}|D) = \frac{1}{T}\sum_{i=1}^{T} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2. \tag{7}$$

To fit the parameters of the neural network one hes to solve the optimization problem

$$\mathbf{w}_j^* = \arg\min_{\mathbf{w} \in \mathbb{W}} \big(\mathcal{L}(\mathbf{w})\big). \tag{8}$$

The algorithm of modeling the dynamic system with Lagrangian is:

1. Find analytical expressions for the kinetic T and the potential V energies.
2. Get the Lagrangian $L = T - V$ in (3).

3. Apply the Euler-Lagrange constraint (1).
4. Solve the resulting system of differential equations.

Instead of the dynamic system, parametrized by a neural network

$$\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}|\mathbf{w}), \quad \text{where} \quad \mathbf{w} - \text{parameters of neural netowork}, \tag{9}$$

To add prior knowledge of the physics of the system we model the Lagrangian:

$$\hat{L} = f(\mathbf{x}|\mathbf{w}), \quad \text{where} \quad \hat{L} - \text{estimation of Lagrangian at point } \mathbf{x}. \tag{10}$$

The key idea is to fit the Lagrangian L by a neural network f, one has to obtain the expression of the Euler-Lagrange constraint and back propagate the error through this constraint:

$$\ddot{\mathbf{q}} = (\nabla_{\dot{\mathbf{q}}\dot{\mathbf{q}}} L)^{-1} \left[\nabla_{\mathbf{q}} L - (\nabla_{\dot{\mathbf{q}}\mathbf{q}} L)\,\dot{\mathbf{q}}\right]. \tag{11}$$

The diagram of the proposed neural network is

$$\mathbf{x}_i = (\mathbf{q}_i, \dot{\mathbf{q}}_i) \xrightarrow{\hat{L}\colon (\mathbf{x}|\mathbf{w}) \to L_{\mathbf{x}}} L_{\mathbf{x}} \xrightarrow{\nabla L} \frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}} \xrightarrow{\ddot{\mathbf{q}}_i = g(\mathbf{q}_i, \dot{\mathbf{q}}_i)} \mathbf{y}_i = \ddot{\mathbf{q}}_i\ . \tag{12}$$

The neural network $\hat{L}\colon (\mathbf{x}|\mathbf{w}) \to L_{\mathbf{x}}$ from (12) is a fully-connected network with three layers with the SoftPlus activation function. One has to estimate $\hat{L}$ of the Lagrangian L, differentiate $\hat{L}$ and obtain $\nabla_{\dot{\mathbf{q}}\dot{\mathbf{q}}}\hat{L}$, $\dot{\mathbf{q}}\hat{L}$, and $\nabla_{\dot{\mathbf{q}}\mathbf{q}}\hat{L}$ at point $\mathbf{x}$. To obtain the acceleration $\hat{\mathbf{y}}$, the values of the gradients are substituted into the Euler-Lagrange equation (1) in the form (11), from which one has to estimate $\hat{\mathbf{y}}$. For the given coordinates $(\mathbf{q}, \dot{\mathbf{q}})$ we have a model with prior knowledge of the energy conservation which obtains the dynamics $(\dot{\mathbf{q}}, \ddot{\mathbf{q}})$.

3 Approximation of the norm in Lagrangian space

We assume the Lagrangians of trajectories are compact. Denote by $\mathbb{L}$ the functional Lagrangian space:

$$\hat{L} = \mathbf{x} \to L_x, \quad g \in \mathbb{L},$$

where $\mathbf{x} \in \mathbb{R}^{2\times r}$ is the vector of generalized coordinates and velocities, $L \in \mathbb{R}$ is the value of the Lagrangian of the dynamic system. The parametric family of functions (9) approximate the Lagrangian

$$\mathbb{L}_\epsilon = \{\hat{L}^{(\epsilon)}\colon (\mathbf{x}|\mathbf{w}) \to \hat{L}(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^{2\times r}, \hat{L}(\mathbf{x}) \in \mathbb{R}\}, \quad \mathbb{L}_\epsilon \subset \mathbb{L},$$

where $\mathbf{w}$ is the vector of parameters of the approximation model, ϵ is the approximation accuracy.

To construct the norm, obtain a system of equations and calculate the accelerations. The Lagrangian formalism models the dynamic system with coordinates $\mathbf{x}_t = (\mathbf{q}, \dot{\mathbf{q}})$. It starts in the state $\mathbf{x}_0$ and ends in the state $\mathbf{x}_1$. Define the action

$$S = \int_{t_0}^{t_1} L dt.$$

The action defines a path from $\mathbf{x}_0$ to $\mathbf{x}_1$ with the coordinates $\mathbf{x}_t$ in the time interval t_0, t_1. The path minimizes the action S. The solution is the Euler-Lagrange equation (1). It defines the system dynamics. Obtain the acceleration of each component $\ddot{\mathbf{q}}$ from the chain rule. Write the time derivative through the coordinate derivatives:

$$\left(\frac{\partial}{\partial \mathbf{q}}\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)\frac{dq}{dt} + \left(\frac{\partial}{\partial \dot{\mathbf{q}}}\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)\frac{d\dot{\mathbf{q}}}{dt} = \frac{\partial L}{\partial \mathbf{q}}.$$

Rewrite the time derivatives for clarity

$$\frac{\partial}{\partial \mathbf{q}} \frac{\partial L}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}} + \frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial L}{\partial \dot{\mathbf{q}}} \ddot{\mathbf{q}} = \frac{\partial L}{\partial \mathbf{q}}.$$

Move the summands so that only the unknowns ones remain on the left side:

$$\frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial L}{\partial \dot{\mathbf{q}}} \ddot{\mathbf{q}} = \frac{\partial L}{\partial \mathbf{q}} - \frac{\partial}{\partial \mathbf{q}} \frac{\partial L}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}}.$$

Divide both parts to get the unknown variable:

$$\ddot{\mathbf{q}} = \left(\frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial L}{\partial \dot{\mathbf{q}}} \right)^{-1} \left[\frac{\partial L}{\partial \mathbf{q}} - \frac{\partial}{\partial \mathbf{q}} \frac{\partial L}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}} \right].$$

Rewrite the partial derivatives with nabla-notation:

$$\ddot{\mathbf{q}} = \left(\nabla_{\dot{\mathbf{q}}\dot{\mathbf{q}}} L \right)^{-1} \left[\nabla_q L - \left(\nabla_{\dot{\mathbf{q}}\mathbf{q}} L \right) \dot{\mathbf{q}} \right], \quad \text{where the Hessian} \quad \mathbf{H} = \nabla_{\dot{\mathbf{q}}\dot{\mathbf{q}}} L = \frac{\partial^2 L}{\partial \dot{\mathbf{q}} \partial \dot{\mathbf{q}}}. \tag{13}$$

The condition $\det \mathbf{H} \neq 0$ at any point must be held since we assume non-degenerate system of linear equations.

$$\mathbf{H}\ddot{\mathbf{q}} = \frac{\partial L}{\partial \mathbf{q}} - \frac{\partial}{\partial \mathbf{q}} \frac{\partial L}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}}, \tag{14}$$

Lemma 1. *If the Lagrangians are set on a compact, then there exist non-negative numbers a, b such that* $\det \mathbf{H} \geq a$ *and the eigenvalues of the matrix* $\mathbf{H}$ *are at least* b.

Proof. Assume the opposite: $\mathbf{H}$ is a continuous function since its values are the second derivatives of L, hence $\det \mathbf{H}$ and its eigenvalues are also continuous and hence reach their minimum and maximum. We obtained contradictions, since the matrix is nondegenerate at all points. □

Since the equations are invariant with respect to Lagrangian multiplication by a constant, we can choose Lagrangians for the systems such that the eigenvalues of the matrix $\mathbf{H}$ are at least 1. Now we can see that a norm can be defined on the space $\mathbb{L}$.

Lemma 2. *The following operator is linear:*

$$A(L) = \frac{\partial L}{\partial \mathbf{q}} - \frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial L}{\partial \dot{\mathbf{q}}} \mathbf{q}.$$

Proof.

$$A(L) = \frac{\partial \alpha L_1 + \beta L_2}{\partial \mathbf{q}} - \frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial \alpha L_1 + \beta L_2}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}} = \frac{\partial \alpha L_1}{\partial \mathbf{q}} + \frac{\partial \beta L_2}{\partial \mathbf{q}} - \left(\frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial \alpha L_1}{\partial \dot{\mathbf{q}}} + \frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial \beta L_2}{\partial \dot{\mathbf{q}}} \right) \dot{\mathbf{q}} =$$
$$\alpha \frac{\partial L_1}{\partial \mathbf{q}} + \beta \frac{L_2}{\partial \mathbf{q}} - \alpha \frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial L_1}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}} - \beta \frac{\partial}{\partial \dot{\mathbf{q}}} \frac{\partial L_2}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}} = \alpha A(L) + \beta A(L).$$

□

Introduce the norm L_2 in the space $\mathbb{L}$. It is semi-norm $\|L\|_L = \|(A(L), \mathbf{H}(L))\|_2$ since the linear operator preserves the absolute homogeneity and triangle inequality.

But the set $\|L\|_L = 0$ is also a subspace of L_2, since $A(L)$, $\mathbf{H}(L)$ are linear operators, and hence their kernels are linear spaces. Then we can introduce a factor space in which the equivalence relation is equality to 0 of the semi-norm $\|L\|_L$. Analyze the introduced equivalence.

Theorem 1. $\|\delta L\|_L = 0 \Leftrightarrow$ *a.e.* $\delta \ddot{q} = 0$, *where* $\delta L = L' - L$, $\delta \ddot{q} = \ddot{q}' - \ddot{q}$, *where* L', $L \in \mathcal{X}$ *are variations of the Lagrangian and acceleration over a set* $\mathcal{X}$, *where functional equation* (14) *respectful to* L *has unique solution in* $\mathcal{X}$, *and* $\forall L \in \mathcal{X} : \ \mathbf{H}(L) \not\equiv 0,.$

Proof. Since the pair $\ddot{q}'$ and $(\ddot{q}')$ is the solution of (14),

$$A(L) = \mathbf{H}(L)\ddot{\mathbf{q}}, \tag{15}$$

and

$$A(L') = \mathbf{H}(L')(\ddot{\mathbf{q}}'). \tag{16}$$

Using $\ddot{q} = \ddot{q}' - \ddot{q}$, $\delta L = L' - L$ rewrite (16) as

$$A(L + \delta L) = \mathbf{H}(L + \delta L)(\ddot{\mathbf{q}} + \delta\ddot{\mathbf{q}}). \tag{17}$$

Expand the right side of equation (17), consider $\mathbf{H}$ and A are the linear functionals

$$\begin{aligned} A(L) + A(\delta L) = (\mathbf{H}(L) + \mathbf{H}(\delta L))(\ddot{\mathbf{q}} + \delta\ddot{\mathbf{q}}) = \\ = \mathbf{H}(L)\ddot{\mathbf{q}} + \mathbf{H}(L)\delta\ddot{\mathbf{q}} + \mathbf{H}(\delta L)\ddot{\mathbf{q}} + \mathbf{H}(\delta L)\delta\ddot{\mathbf{q}}. \end{aligned} \tag{18}$$

Subtract the expression (15) from the expression (18),

$$A(\delta L) = \mathbf{H}(\delta L)\ddot{\mathbf{q}} + \mathbf{H}(L)\delta\ddot{\mathbf{q}} + \mathbf{H}(\delta L)\delta\ddot{\mathbf{q}}. \tag{19}$$

If $|(A(\delta L), \mathbf{H}(\delta L))\|_2 = 0$, then a.e. $A(\delta L) = 0$, $\mathbf{H}(L) = 0$. Using $A(\delta L) = 0$, $\mathbf{H}(L) = 0$ rewrite the equation (19) as

$$0 = \mathbf{H}(L)\delta\ddot{\mathbf{q}}. \tag{20}$$

As $\mathbf{H}(L) \not\equiv 0$, then a.e. $\ddot{\mathbf{q}} = 0$.

If a.e. $\ddot{\mathbf{q}} = 0$, then a.e. rewrite the equation (19)

$$A(\delta L) = \mathbf{H}(\delta L)\ddot{\mathbf{q}}. \tag{21}$$

Using $\ddot{q} = \ddot{q}' - \ddot{q}$, $\delta L = L' - L$ and considering $\mathbf{H}$ and A are the linear functionals rewrite (21) as

$$A(L' - L) = \mathbf{H}(L' - L)\ddot{\mathbf{q}} \tag{22}$$

$$A(L') - A(L) = \mathbf{H}(L')\ddot{\mathbf{q}} - \mathbf{H}(L)\ddot{\mathbf{q}} \tag{23}$$

Add the expression (15) to the expression (23)

$$A(L') = \mathbf{H}(L')\ddot{\mathbf{q}}\ddot{\mathbf{q}} \tag{24}$$

As $L' \in \mathcal{X}$ functional equation (14) respectful to L has unique solution, then a.e. $A(L') = A(L)$, $\mathbf{H}(L') = \mathbf{H}(L)$

$$\text{a.e.} A(L') = A(L), \mathbf{H}(L') = \mathbf{H}(L) \to \|(A(\delta L), \mathbf{H}(\delta L))\|_2 = 0 \tag{25}$$

□

Note that in most cases $\|\mathbf{H}(\delta L)\|_2 = 0$, because it is always possible to choose the coordinates so, by means of the Legendre transform, that the Hesse matrix is unitary, for this purpose it is necessary to pass to the canonical coordinates, and thus it is enough to compare Classification of trajectories of dynamic systems using physically-informed neural networks 13 only the first part of the norm $\|A(L)\|_2$.

This norm cannot be calculated exactly in the general case, because for this purpose it is necessary to calculate the integral of the function on the whole numerical line, but it can be approximated

By definition of L_2-norm

$$\|(A(L), \mathbf{H}(L))\|_2 = \sqrt{\int\!\!\int |A(L)\,(\mathbf{q}, \dot{\mathbf{q}})\,|^2 + \|\mathbf{H}(L)\,(\mathbf{q}, \dot{\mathbf{q}})\,\|_2^2 d\Omega} \tag{26}$$

If integral (26) is defined, then Riemann sum's converge to integral, when cell shrinks to zero, $\|x\| \to 0$

$$\begin{gathered}\sum_{\substack{i_1=1,\ldots,i_n=1\\ i=(i_1,\ldots,i_n)}}^{N,\ldots,N} |A(L)\,(\mathbf{q}_i, \dot{\mathbf{q}}_i)\,|^2 + \|\mathbf{H}(L)\,(\mathbf{q}_i, \dot{\mathbf{q}}_i)\,\|_2^2)\frac{\Delta X_1 \cdot \ldots \cdot X_n}{N \cdot \ldots . \cdot N} \\ \overset{N\to\infty}{\longrightarrow} \|(A(L), \mathbf{H}(L))|\Omega\|_2 \overset{\mu(\Omega)\to\infty}{\longrightarrow} \|(A(L), \mathbf{H}(L))\|_2,\end{gathered} \tag{27}$$

where cells defined on uniform grid over $\Omega = [-\Delta X_1/2, \Delta X_1/2] \times \ldots \times [-\Delta X_n/2, \Delta X_n/2]$ and $(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ are coordinates of centers of cells.

Rewrite expression (27) using the fact $\mu(\Omega) = \Delta X_1 \cdot \ldots \cdot X_n$

$$\begin{gathered}\sum_{\substack{i_1=1,\ldots,i_n=1\\ i=(i_1,\ldots,i_n)}}^{N,\ldots,N} |A(L)\,(\mathbf{q}_i, \dot{\mathbf{q}}_i)\,|^2 + \|\mathbf{H}(L)\,(\mathbf{q}_i, \dot{\mathbf{q}}_i)\,\|_2^2)\frac{\Delta X_1 \cdot \ldots \cdot X_n}{N \cdot \ldots . \cdot N} = \\ \mu(\Omega) \cdot \sum_{i_1=1,\ldots,i_n=1}^{N,\ldots,N} \frac{A(L)\,(\mathbf{q}, \dot{\mathbf{q}})^2 + \|\mathbf{H}(L)\,(\mathbf{q}, \dot{\mathbf{q}})\,\|_2^2}{N \cdot \ldots . \cdot N} = \\ \mu(\Omega) \cdot \overline{|A(L)\,(\mathbf{q}, \dot{\mathbf{q}})\,|^2 + \|\mathbf{H}(L)\,(\mathbf{q}, \dot{\mathbf{q}})\,\|_2^2}.\end{gathered} \tag{28}$$

Since we are solving a classification problem, let us compare to each Lagrangian L_i its class $\mathcal{A}_k = \{L_i | y_i = k\}$, where y_i is the class label of Lagrangian. Thus, we have a finite family of non-intersecting closed closed singular sets $\mathcal{A}$ in the space $\mathbb{L}$ of Lagrangians, where each family represents a class corresponding to a human motion.

Theorem 2. *Let there be a finite family of non-intersecting closed convex sets $\mathcal{A}$ in the normalized space $\mathbb{L}$, then there exists $\epsilon > 0$ such that for any transformation of the space ϕ such that*

$$\|\phi(\mathcal{A}_i) - \mathcal{A}_i\| < \epsilon$$

sets from the family $\phi\mathcal{A}$ is pairwise strongly separable.

Proof. According to the first Hahn-Banach separability theorem, sets from the family $\mathcal{A}$ are pairwise separable

$$\underset{x\in\mathcal{A}_i}{\langle a, x\rangle} \le \gamma_i < \gamma_j \le \underset{y\in\mathcal{A}_j}{\langle a, y\rangle}.$$

Since the functional a is continuous, for any $\epsilon > 0$ there exists a neighborhood of zero $U(0)$ such that

$$\underset{y\in U(0)}{\langle a, y\rangle} \le \epsilon.$$

Let's take ϵ_{ij} for each pair of classes i, j such as

$$\epsilon_{ij} = \frac{|\gamma_i - \gamma_j|}{3},$$

then

$$\underset{x\in\mathcal{A}_i+U(0)}{\langle a, x\rangle} \le \gamma_i + \epsilon < \gamma_j + \epsilon \le \underset{y\in\mathcal{A}_j+U(0)}{\langle a, y\rangle}.$$

If we take $\epsilon = \min\limits_{i,j} \epsilon_{ij}$, then the sets $\mathcal{A}_i$ are pairwise strongly separable. □

This theorem allows us to study the properties of Lagrangians on their approximations, which simplifies the problem markedly.

Based on this theorem, we can state a sufficient property for the approximation algorithm. There must exist $\epsilon > 0$ such that for any $\mathcal{L} \in \mathbb{L}$ it is true that $\|\mathcal{L} - \mathcal{L}_\epsilon\| < \epsilon$. In other words, it is necessary that the distance from the approximated Lagrangian to the true one was small. I.e., it is required that the neural network does not overtrain and approximates the Lagrangian with sufficient accuracy. This condition is fulfilled for a correctly constructed network.

From Theorem 2 follows the first method of classifying the Lagrangian. Since we have introduced a norm in the space $\mathbb{L}$, it generates a metric. Hence, we can use metric methods of classification. In this paper, we propose to use the parameters of a neural network approximating the Lagrangian for classification. The following theorem helps us in this.

Theorem 3. *Let there be a finite family of sets $\mathcal{A}$ in a normalized space $\mathbb{L}$, $\epsilon > 0$, and a transformation of the space ϕ such that $\|\phi(\mathcal{A}_i) - \mathcal{A}_i\| < \epsilon$ sets from the family $\phi\mathcal{A}$ are pairwise strongly separable, and a mutually unambiguous continuous mapping $\psi : \mathbb{L}_\epsilon \to \mathbb{R}^n$ the inverse mapping is continuous and convex. Then the sets $\mathcal{A}_i$ are pairwise strongly separable.*

Proof. Let us immerse in the space $\mathbb{R}^n$. In it $\phi\psi\mathcal{A}_i$ are pairwise strongly separable. Then each set is bounded by the corresponding hyperplanes

$$\underset{x \in \phi\psi\mathcal{A}_i}{\langle a, x \rangle} \le \gamma_1.$$

Let us extend each set to hyperplanes and consider the already extended ones. Each of these sets is compact. Since ψ^{-1} is convex and continuous, the image of the extended sets is also convex and compact. Hence, the extended sets are the pairs are highly separable. □

The theorem gives sufficient conditions for the compactness hypothesis to hold for initial sets. But its conditions are quite strong. It requires that the weights be continuous, which may not be the case. This method is convenient because you don't have to use many features, but it is not suitable for proving that the hypothesis is not fulfilled for a data set. If it is not fulfilled, it is better to use methods based on the metric given by the norm.

4 Classification problem statement for trajectories

For the given sample set (5) we define the classification model $p(y|X, D)$ for $y \in \{1, \dots, K\}$, the set of class labels for the trajectories. The classification quality is defined by

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} [\hat{y}_i = y_i].$$

Despite this formulation of the problem, it is usually physically justified to classify dynamical systems, which in the case under consideration are identified with the Lagrangians L, rather than the realizations of the systems themselves in the form of trajectories.

Therefore, the paper considers a slightly different, but in this interpretation physically equivalent formulation of the problem.

We need to find a classification method $p(\hat{y}|X, D) = p(\hat{y}|L, D)p(L|X)$, where $X = \{(X_i)\}_{i=1}^N$ is a set of length N of trajectories of dimension r and length t, D is the given training sample, $\hat{y} \in \{1, \dots, K\}$ are the predicted class labels, and $p(L|\mathbf{X})$ matches the trajectory $\mathbf{X}$ to the Lagrangian L.

In this paper, as a method $p(L|X) = [L = \hat{L}]$ we consider point-wise estimation of the trajectory $\mathbf{X}$ by a parameterized estimate of the Lagrangian $\hat{L}$ by a Lagrangian neural network from the considered regression problem for time series. But such a formulation makes it possible to use the probabilistic estimator $p(L|\mathbf{X})$ for the Lagrangian L.

5 Loss function

Similar to the problem with discretization of continuous space of HNN [17]. The solution of equation (14) is invariant with respect to $\ddot{q}$, where L is multiplied by $c > 0$. In this case, when L tends to zero, the error $\delta\ddot{q}$ tends to infinity. It also means that the solution of the optimization problem (8) is essentially ambiguous. For regularization of the problem impose restrictions on the found solution

$$\lambda\left(\mathbf{H}\right) > 1. \tag{29}$$

In accordance with Lemma 1, the solution on the set with constraint (29) is not worse than for the problem without constraints. At the same time, this constraint does not solve the problem with ambiguity of the problem. This requires a modification of the loss function. We add a summand equal to the inconsistency in the solution of SLAE (14) with respect to $\ddot{q}$

$$\mathbf{H}\ddot{\mathbf{q}} = b,$$

$$\mathbf{H}\hat{\ddot{\mathbf{q}}} = \hat{b} = \left(\nabla_{\dot{\mathbf{q}}\dot{\mathbf{q}}}L\right)^{-1}\left[\nabla_{q}L - \left(\nabla_{\dot{\mathbf{q}}\mathbf{q}}L\right)\dot{\mathbf{q}}\right],$$

$$\mathcal{L}_{\text{add}}(\mathbf{w}) = \frac{1}{T}\sum_{i=1}^{T}\|\hat{\mathbf{b}}_i - \mathbf{b}_i\|_2^2. \tag{30}$$

According to Lemma 1 for $\hat{L}$ an exact estimate $\ddot{\mathbf{q}}$ gives $\mathcal{L}_{\text{add}} = 0$. Moreover, if the constraints (29) are satisfied, $\mathcal{L}_{\text{add}} > \mathcal{L}$ and we can replace the loss function by $\mathcal{L}$. Thus, we obtained the following loss function, taking into account the constraints

$$\mathcal{L} = \mathcal{L}_{\text{add}} + \mathcal{L}_{\text{restriction}}, \tag{31}$$

$$\mathcal{L}_{\text{add}}(\mathbf{w}) = \frac{1}{T}\sum_{i=1}^{T}\|\hat{\mathbf{b}}_i - \mathbf{b}_i\|_2^2, \tag{32}$$

$$\mathcal{L}_{\text{restriction}}(\mathbf{w}) = +\infty[(\mathbf{H}) > 1]. \tag{33}$$

6 Classification method

The classification method has the following form

$$p(\hat{y}|X, D) = p(\hat{y}|L, D)p(L|X).$$

As mentioned in Section 4, the method $p(L|X)$ assigns to each trajectory an estimate of the Lagrangian $\hat{L}$ from the linear regression problem for a Lagrangian neural network.

But it is not possible to simply apply the vector method of classifying L as $p(\hat{y}|L, D)$ since L is a vector of an infinite-dimensional space $\mathbb{L}$. For this purpose, the space L is projected onto the finite-dimensional space in such a way that the norm of the vectors under consideration changes by no more than a predetermined value $\epsilon > 0$.

As norm in space $\mathbb{L}$ is taken norm $\|L\|_L$. To its approximation on a finite set of points is used to calculate the norm

$$\|L\|_L \approx \overline{|A(L)\left(\mathbf{q}, \dot{\mathbf{q}}\right)|^2 + \|\mathbf{H}(L)\left(\mathbf{q}, \dot{\mathbf{q}}\right)\|_2^2}.$$

If we take the values of

$$A(L)\left(\mathbf{q}, \dot{\mathbf{q}}\right) \quad \text{and} \quad \|\mathbf{H}(L)\left(\mathbf{q}, \dot{\mathbf{q}}\right)$$

as the value of the vector, we obtain the projection of the space L onto a finite-dimensional space. The projection is the input to the classifier. This projection is fed to the input of the classifier. Thus we obtain the classification method:

$$\begin{tikzcd}
X \arrow[r, "LNN"] & \hat{L} = g(x|w) \arrow[rr, "L_N^i = A(L)(\mathbf{q}_i, \dot{\mathbf{q}}_i)"] & & L_N \in \mathbb{R}^N \arrow[r, "\text{classifier:} \quad \mathbb{R}^N \to K"] & \hat{y} \\
& Q_N \sim U([-L, L]^{2r}) \arrow[ur] & & &
\end{tikzcd} \tag{34}$$

At the beginning, during the training phase, a set of points

$$Q_N \sim U([-L, L]^{2r}), |Q_N| = N$$

in which the function values are sampled is selected in advance. For trajectory X estimate of $\hat{L}$ is found by Lagrangian neural network. Then for function $\hat{L}$ is searched for the values of $A(L)\left(\mathbf{q}, \dot{\mathbf{q}}\right)$ and $\|\mathbf{H}(L)\left(\mathbf{q}, \dot{\mathbf{q}}\right)$ at selected points Q_N. From these values a vector L_N is collected. Then this vector L_N is fed to the input of the vector classification method. Thus, the points Q_N act as hyperparameters, and the trained part of the model is the classifier $C : L_N \to \hat{y}$.

7 Computational experiment

The aim of the experiment is to classify multivariate time series represented as trajectories of physical systems. As subtasks, it is necessary to confirm the hypothesis of compactness of classes, to confirm in practice the theory that the Lagrangian space can be normalized by the proposed norm and sampled by a predetermined set of points, and to construct a classifier using it.

The computational experiment used the Physical Activity Monitoring (PAMAP2) dataset [16]. It contains recordings from three sets of gyroscopes and accelerometers, attached to the wrist of the dominant arm, attached to the chest, attached to the elbow of the predominant hand. Number of subjects: $M = 9.$. Number of activity types(classes): $K = 24$, each activity lasted 2-4 minutes [16], sampling frequency 100 Hz.

Of all the sensors presented in the dataset, gyroscopes were selected as the most suitable for classification. walking, running, cycling and non-moving state. Data processing consists of recovering the gaps in the data, associated with the absence of signals from gyroscopes, as well as obtaining the values of angles and angular accelerations from angular velocities obtained from gyroscopes.

Some data in the rows is lost due to communication problems. To recover the series, spline interpolation is used to obtain quadratic accuracy from the grid size. Numerical differentiation and integration are used to recover acceleration $\ddot{\mathbf{q}}$, velocity $\dot{\mathbf{q}}$ and coordinates $\mathbf{q}$. We use the central difference for differentiation

$$\ddot{\mathbf{q}}_k = \frac{\dot{\mathbf{q}}_{k+1} - \dot{\mathbf{q}}_{k-1}}{2h}, \quad |E(\ddot{q}_k)| \leq \frac{h^2}{6}\mathbf{q}^{(4)},$$

and the Cortes formula for integration

$$\mathbf{q}_k = \int_0^{t_k} \dot{\mathbf{q}}(t) d\mathbf{x} \approx \frac{h}{3} \sum_{k=1}^{N-1} \left(\dot{\mathbf{q}}_{k-1} + 4\dot{\mathbf{q}}_k + \dot{\mathbf{q}}_{k+1}\right), \quad |E(\mathbf{q}_k)| \leq \frac{(b-a)}{2880} h^4 \max \left|\mathbf{q}^{(4)}\left(\mathbf{x}\right)\right|.$$

The resulting trajectories are passed as objects $\mathbf{X}_i = (\mathbf{q}, \dot{\mathbf{q}})(t)_i$ and features $\mathbf{Y}_i = \ddot{\mathbf{q}}(t)_i$. The neural network presented in [3] is trained on the i-th trajectory separately and an estimate of the Lagrangian $\hat{L}_i$, mapped to the i-th trajectory is obtained. his Lagrangian, defines the dynamical system and allows to recover the acceleration $\ddot{\mathbf{q}}(t)_i$. This yields a set of objects $\hat{L}_i$, and a set of labels y_i, where y_i is the trajectory class X_i.

As a visualization and evaluation of trajectory recovery with the necessary precision, LNN predictions of acceleration $\ddot{\mathbf{q}}(t)$ for trajectories $(\mathbf{q}, \dot{\mathbf{q}})(t)$ were constructed and compared with the real ones.

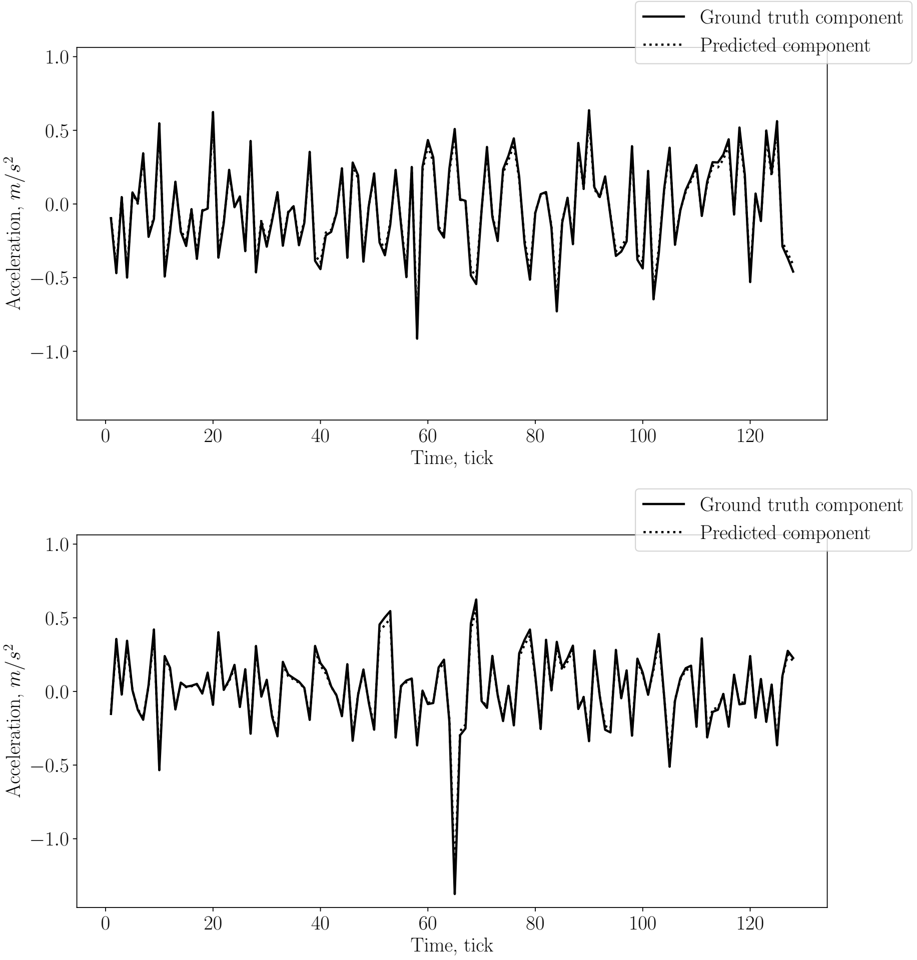

Figure 2 Time series of acceleration versus time for the test sample (a) the first component, (b) the second component

Figures 2 and 3 show that the trajectories are exactly reconstructed by their Lagrangian, and we can predict them over a sufficiently large time interval, the losses on the test and training samples are close, and the trajectories retain their appearance. This means that, according to the assumption, the Lagrangian space is a hidden space from which all trajectories are generated.

Thus $\hat{L}_i$ vectorizes the X_i data for further classification. In the remark to lemma 2, a norm on the Lagrangian space is introduced, which represents it as a Hilbert space. Finally, the vectorization is taken from Remark 2 in which the Lagrangian space is projected onto the Euclidean space $\mathbb{R}^n$. The vectorization of a trajectory corresponds to the vectorization of their Lagrangians.

In contrast to [3], an additional regularization of the loss function is proposed. Since in the regions where the eigenvalues of the Hessian

$$\mathbf{H} = (\nabla_{\dot{\mathbf{q}}\dot{\mathbf{q}}} L)_{ij} = \frac{\partial^2 L}{\partial \dot{q}_j \partial \dot{q}_i}$$

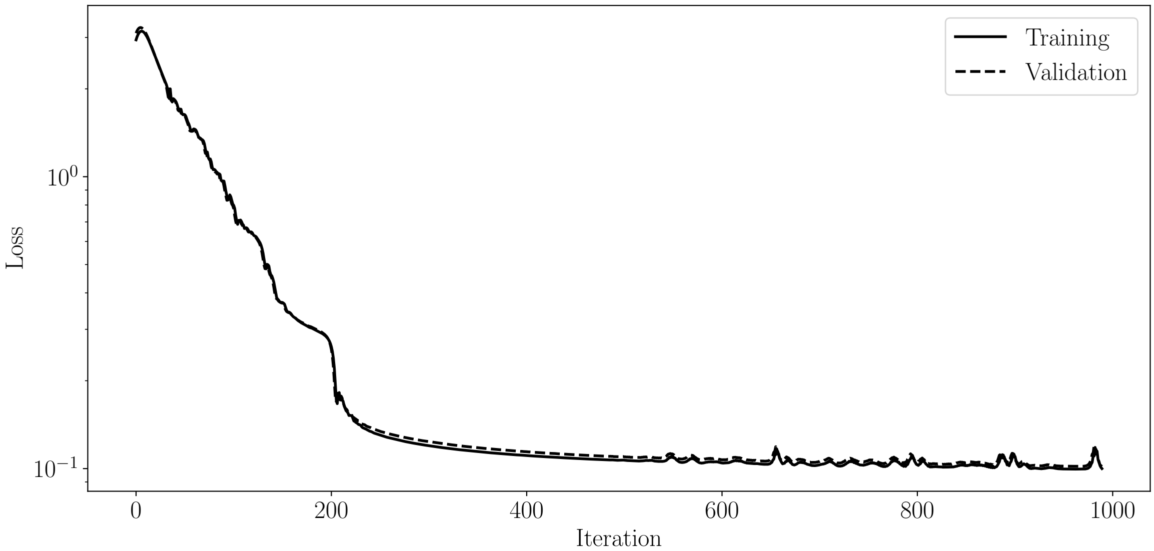

Figure 3 The learning curve shown the optimization of the Lagrangian reconstruction

tend to 0 the point of equilibrium is unstable, and the trajectories show nonphysical oscillations. Therefore, we introduce an additional term

$$\mathcal{L}_{\text{restriction}} = \sum \alpha a(\beta(1-\lambda(\mathbf{H})_i),$$

where a is some linear activation function, the best fit was $a = \text{SiLU} = x\sigma(x)$. The hyperparameters α, β are adjusted depending on the type of data, for this dataset $\alpha = 5, \beta = 12$ are selected.

For the resulting Lagrangian estimator, we can sample the function values in any point $(\mathbf{q}, \dot{\mathbf{q}})$. Accordingly, the norm on the basis of which the metric is constructed is sampled, and based on this metric, the underlying algorithms classify the trajectories.

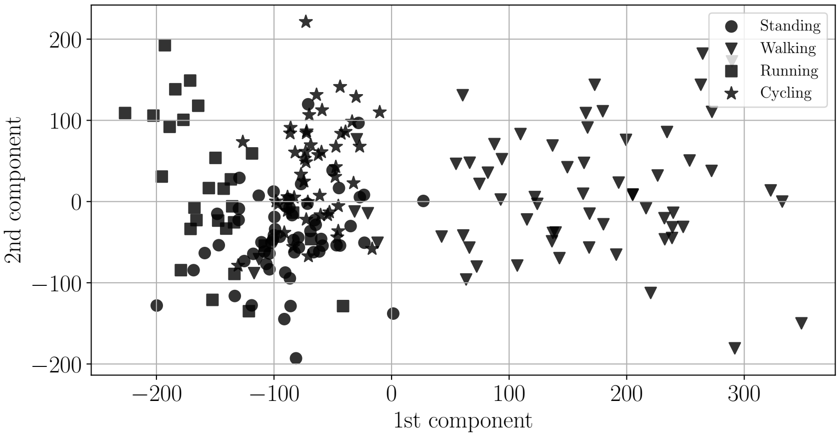

Figure 4 Projection of the Lagrangian estimates of $\hat{L}$ on a 2D-surface

Therefore, we first obtain a set of points at which the Lagrangian values are obtained. This set is obtained before classification, and it is the same for all trajectories. Therefore, it is a hyperparameter for the classifier. We obtain the set $S = \{s_i\}$ of $N = 20000$ points $(\mathbf{q}, \dot{\mathbf{q}})$ uniformly distributed on a cube $-10 \leq x_i \leq 10$. Size the cube is chosen so that it covers the points on which the Lagrangians are trained. The number of points for sampling should be chosen so that the estimate of the norm varies weakly compared to its estimate.

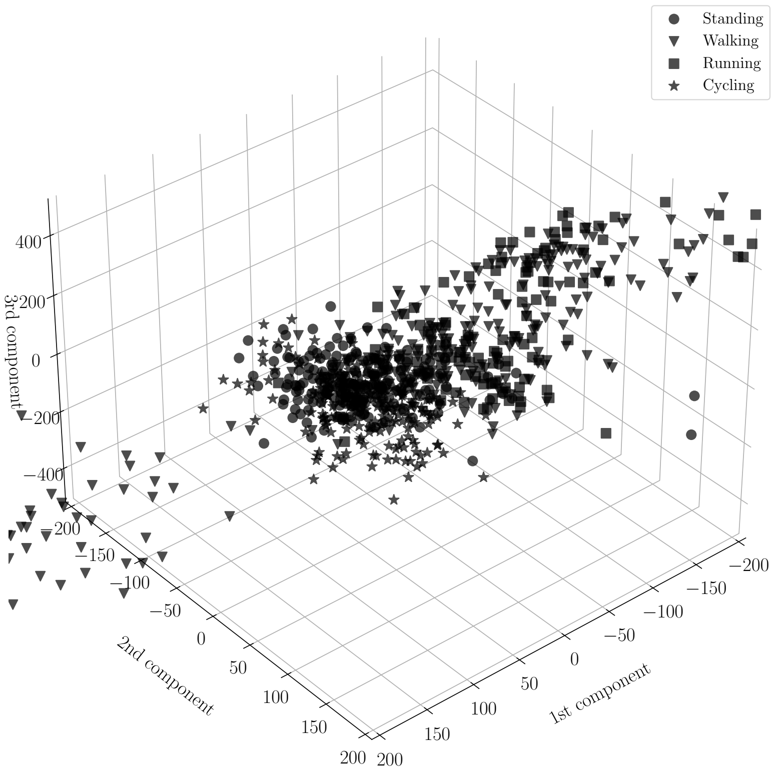

Figure 5 Projection of the Lagrangian estimates of $\hat{L}$ on a 3D-surface

For each object, the values $X_i = \{AL(s_i)\}$ set of its features are obtained. The norm in this space corresponds to the usual Euclidean norm, and the basic algorithms of veto classification are used to classify objects in this space. We propose to classify objects by their features, i.e., sampled norm values.

Table 1 Result of classifiers on the proposed data vectorization

Classifier	Metric		
	Accuracy	Balanced-accuracy	F1-macro
Logistic regression	0.927 ± 0.014	0.924 ± 0.13	0.927 ± 0.14
Gaussian process	$\mathbf{0.946 \pm 0.010}$	$\mathbf{0.941 \pm 0.011}$	$\mathbf{0.946 \pm 0.010}$
Random forest	0.932 ± 0.007	0.928 ± 0.008	0.933 ± 0.008
K-nearest neighbors	0.939 ± 0.009	0.935 ± 0.010	0.940 ± 0.008
SVC with Gaussian kernel	0.933 ± 0.012	0.927 ± 0.013	0.933 ± 0.011

To regularize the model and reduce dimensionality, we apply Principal Component Analysis and select the first three principal components in 2D and 3D plots in the Figures 4 and 5. The data are separated into clusters. They visually fit the normal distribution so we use the Gaussian kernel in metric classifiers. The clusters are separated by linear surfaces in the presented Lagrangian space.

The Random forest classifier was chosen for the study because it works well with any complex datasets, linear regression and Gaussian process as simple enough classifiers that learn quickly and therefore it is useful to know if they work on a dataset, and SVC with a Gaussian kernel because visual analysis assumes that the data are normally distributed.

The Gaussian process classifier shows the best results according to all metrics in comparison to the rest of the models. The Metric classifiers, SVC and K-nearest neighbors, show similar results. Surprisingly, the logistic regression classifier recedes in accuracy, when the number of components

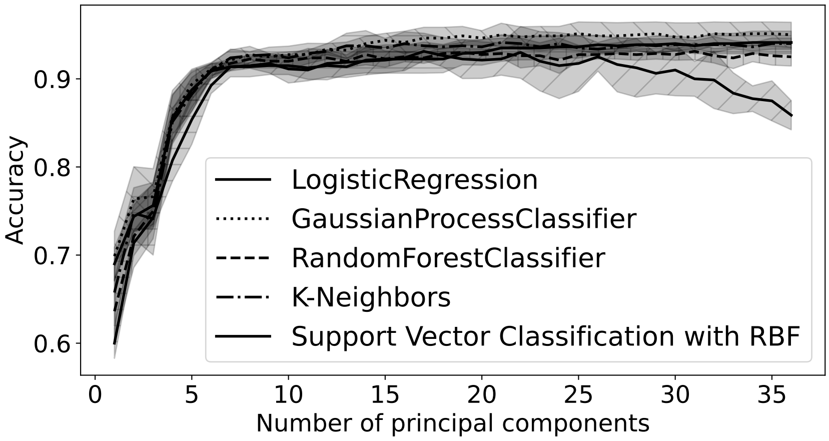

Figure 6 Classification accuracy of the selected methods depending on the number of principal components

is bigger than 25. Among the the widely-used time series classification methods, the random forest classifier, TimeSeriesForestClassifier from the Sktime library shows the accuracy 78%. It is much worse than the LNN classifier. So, we select the Gaussian process classifier.

8 Conclusion

The method provides a convenient way to project the data onto a plane for further visualization. Visual analysis of the plots shows that the represented Lagrangian space corresponds to the space of features that define a class of trajectories, i.e., they determine the type of motion. Similar results are shown by training linear models on the given vector space, the graphs show that linear models separate the classes represented in the dataset with an accuracy of 0.95. At the same time, the proposed vectorization is time-consuming and requires tuning the regularization parameters to achieve the necessary approximation accuracy on specific sets of trajectories.

Approximation of trajectories with dynamic system Lagrangians is a powerful tool for trajectory classification. It has many advantages over classification methods which do not use information about the physics of the system. This method is multivariate, there is no need to make assumptions about the relationship between different components of trajectories. It is robust to changes in the data due to the proposed way of Lagrangian approximation. The proposed time series representation preserves assumptions about the physical nature of the signal measurements.

9 Declarations

Data availability: Not Applicable.
Conflict of interest: The authors declare they have no financial or personal relationship with any third party whose interests could be positively or negatively influenced by the article's content.
Research Involving: Human and/or Animals Not applicable.
Informed Consent: Not applicable.

References

[1] Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nature Reviews Physics **3**(6), 422–440 (2021) https://doi.org/10.1038/s42254-021-00314-5

[2] Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural

Information Processing Systems, vol. 32 (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/26cd8ecadce0d4efd6cc8a8725cbd1f8-Paper.pdf

[3] Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., Ho, S.: Lagrangian neural networks. In: ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations (2019). https://openreview.net/forum?id=iE8tFa4Nq

[4] Tucker, K., Rege, A.K., Smith, C., Monteleoni, C., Albash, T.: Hamiltonian Learning using Machine Learning Models Trained with Continuous Measurements. arXiv (2024). https://doi.org/10.48550/ARXIV.2404.05526

[5] Bhattoo, R., Ranu, S., Krishnan, N.M.A.: Learning the dynamics of particle-based systems with lagrangian graph neural networks. Machine Learning: Science and Technology **4**(1), 015003 (2023) https://doi.org/10.1088/2632-2153/acb03e

[6] Wessels, H., Weißenfels, C., Wriggers, P.: The neural particle method – an updated lagrangian physics informed neural network for computational fluid dynamics. Computer Methods in Applied Mechanics and Engineering **368**, 113127 (2020) https://doi.org/10.1016/j.cma.2020.113127

[7] Gruber, A., Tezaur, I.: Canonical and noncanonical hamiltonian operator inference. Computer Methods in Applied Mechanics and Engineering **416**, 116334 (2023) https://doi.org/10.1016/j.cma.2023.116334

[8] Olivieri, M., Karakonstantis, X., Pezzoli, M., Antonacci, F., Sarti, A., Fernandez-Grande, E.: Physics-Informed Neural Network for Volumetric Sound field Reconstruction of Speech Signals. arXiv (2024). https://doi.org/10.48550/ARXIV.2403.09524

[9] Han, C.-D., Glaz, B., Haile, M., Lai, Y.-C.: Adaptable hamiltonian neural networks. Physical Review Research **3**(2), 023156 (2021) https://doi.org/10.1103/physrevresearch.3.023156

[10] Zakwan, M., Xu, L., Ferrari-Trecate, G.: Robust classification using contractive hamiltonian neural odes. IEEE Control Systems Letters **7**, 145–150 (2023) https://doi.org/10.1109/lcsys.2022.3186959

[11] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A.: Deep learning for time series classification: a review. Data Mining and Knowledge Discovery **33**(4), 917–963 (2019) https://doi.org/10.1007/s10618-019-00619-1

[12] Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M., Bagnall, A.: The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery **35**(2), 401–449 (2020) https://doi.org/10.1007/s10618-020-00727-3

[13] Rahman, M.M., Farahani, M.A., Wuest, T.: Multivariate time-series classification of critical events from industrial drying hopper operations: A deep learning approach. Journal of Manufacturing and Materials Processing **7**(5), 164 (2023) https://doi.org/10.3390/jmmp7050164

[14] Seto, S., Zhang, W., Zhou, Y.: Multivariate time series classification using dynamic time warping template selection for human activity recognition. In: 2015 IEEE Symposium Series on Computational Intelligence, pp. 1399–1406 (2015). https://doi.org/10.1109/SSCI.2015.199

[15] Dempster, A., Petitjean, F., Webb, G.I.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data Mining and Knowledge Discovery **34**(5), 1454–1495 (2020) https://doi.org/10.1007/s10618-020-00701-z

[16] Reiss, A.: PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository (2012). https://doi.org/10.24432/C5NW2H

[17] Galimberti, C.L., Furieri, L., Xu, L., Ferrari-Trecate, G.: Hamiltonian deep neural networks guaranteeing nonvanishing gradients by design. IEEE Transactions on Automatic Control **68**(5), 3155–3162 (2023) https://doi.org/10.1109/tac.2023.3239430