npj computational materials

www.nature.com/npjcompumats

ARTICLE OPEN

Functional data-driven framework for fast forecasting of electrode slurry rheology simulated by molecular dynamics

Marc Duquesnoy[1,2], Teo Lombardo[1,3], Fernando Caro[1,3], Florent Haudiquez[1], Alain C. Ngandjong[1,3], Jiahui Xu[1,3], Hassan Oularbi[1,3] and Alejandro A. Franco[1,2,3,4]✉

The computational simulation of the manufacturing process of lithium-ion battery composite electrodes based on mechanistic models allows capturing the influence of manufacturing parameters on electrode properties. However, ensuring that these properties match with experimental data is typically computationally expensive. In this work, we tackled this costly procedure by proposing a functional data-driven framework, aiming first to retrieve the early numerical values calculated from a molecular dynamics simulation to predict if the observable being calculated is prone to match with our range of experimental values, and in a second step, recover additional values of the ongoing simulation to predict its final result. We demonstrated this approach in the context of the calculation of electrode slurries viscosities. We report that for various electrode chemistries, the expected mechanistic simulation results can be obtained 11 times faster with respect to the complete simulations, while being accurate with a R^2_{score} equals to 0.96.

npj Computational Materials (2022)8:161 ; https://doi.org/10.1038/s41524-022-00819-2

INTRODUCTION

In a modern world where Artificial Intelligence (AI) and Machine Learning (ML) applications are blooming[1,2], one may think that the use of more traditional mechanistic models based on mathematical descriptions of physical processes is becoming obsolete. However, this is not true since mechanistic models still represent nowadays essential tools to support the analysis of complex systems. In contrast to empirical models (and per se ML approaches), which study the reality to develop equations that are able of describing such reality, mechanistic models are based on theoretical knowledge to predict real-world behavior. Models of this kind are omnipresent in numerous domains such as medicine[3], energy storage[4,5], nanotechnology[6], biology[7,8], and environmental sciences[9]. The significant progress in computational hardware achieved in the last decades boosted the emergence of a massive amount of academic and commercial software[10,11], allowing to solve the equations behind mechanistic models describing systems with increasing complexity[12,13]. One of the application fields for such models is the manufacturing process of lithium-ion batteries (LIBs)[14]. This process encompasses multiple steps and parameters which are interlinked (Figure 1)[15,16]. Such steps concern the slurry preparation, its coating and drying, the calendering of the resulting electrode, the cell assembly, the electrolyte infiltration, and the formation of the solid electrolyte interphase[17–20]. This complex process has been historically simulated using empirical models with parameters fitted considering experimental trends or by using mechanistic models based, for instance, on the continuum fluid dynamics (CFD) approach[21]. In recent years, mechanistic models allowing to predict the electrode microstructure in 3D, as a function of the manufacturing parameters, started to emerge, in particular through our contributions[22,23]. These 3D mechanistic models account for the explicit spatial location and trajectory of active material (AM) and carbon-binder domain (CBD) phases (Figure 1)[4,24,25]. As an example of what happens during the slurry and the drying steps, these models are supported by the coarse-grained molecular dynamics (CGMD) approach accounting for attractive and repulsive forces (force fields (FFs)) between the considered particles[5]. FFs enable to correctly set the CGMD mechanistic model, which can track the spacial location evolution of particles, facilitating the analysis of electrode properties with time. For the case of the calendering step, mechanistic models are supported by the Discrete Element Method (DEM). Given the different steps and dedicated models, it appears meaningful to concatenate these models into a single calculation loop to support the modeling of the full manufacturing process, allowing to evaluate various properties as a function of the manufacturing parameters in a straightforward way[26].

Despite the continuous improvement in the numerical methods used to closely match the experimental data, the execution of these methods is usually time and resources consuming. The overall time and resources required to recover the complete simulation process determine the computational cost of the corresponding simulation and vary as a function of the specific model handled. A high computational cost narrows the number of possible parameter sets that is feasible to consider for testing other conditions[27], preventing the use of mechanistic models in a high-throughput way[28–31]. In the battery manufacturing context, this is especially true since the mechanistic models have a significant computational cost for the generation of 3D-electrode microstructures, but they still need to be run in order to generate the high-fidelity data required for the mechanistic model's validation by comparison with experimental observables needed for the correct setting of FFs. One of these observables is the slurry viscosity as a function of the applied shear rate, which is sensitive to various of the properties of the powders such as the

[1]Laboratoire de Réactivité et Chimie des Solides (LRCS), UMR CNRS 7314, Université de Picardie Jules Verne, Hub de l'Energie, 15, rue Baudelocque, 80039 Amiens, Cedex, France. [2]ALISTORE-European Research Institute, FR CNRS 3104, Hub de l'Energie, 15, rue Baudelocque, 80039 Amiens, Cedex, France. [3]Reseau sur le Stockage Electrochimique de l'Energie (RS2E), FR CNRS 3459, Hub de l'Energie, 15, rue Baudelocque, 80039 Amiens, Cedex, France. [4]Institut Universitaire de France, 103 Boulevard Saint Michel, 75005 Paris, France. ✉email: alejandro.franco@u-picardie.fr

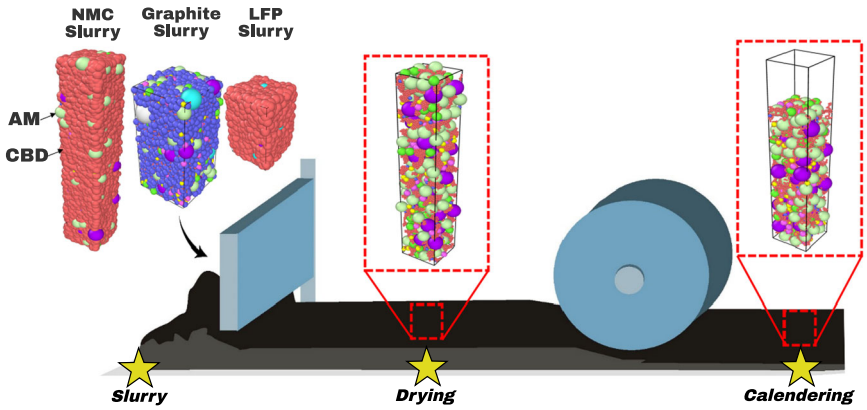

Fig. 1 Schematic of the first steps in the lithium-ion battery electrode manufacturing process which illustrates 3D-generated microstructures resulting from their mechanistic computational modeling. Our modeling methodology focuses first on the electrode slurry and can describe slurries made with different active material chemistries: $LiNi_{0.33}Mn_{0.33}Co_{0.33}O_2$ (NMC-111), Graphite, and $LiFePO_4$ (LFP).

particle size distribution of the AM, the weight ratio between AM and carbon-binder, as well as the solid content in the slurry. Once the slurry microstructure is predicted by the CGMD model, non-equilibrium molecular dynamics (NEMD) simulations are performed to calculate the shear-viscosity curve ($\dot{\gamma}$-η curve) which is compared with the experimental one (see the Supplementary Discussion and Supplementary Figure 1 for the explanations of the working principles of the NEMD approach). The execution time of each calculation must be long enough to achieve convergence and reach a stable viscosity value, usually leading to high overall computational costs.

However, ML techniques can be easily combined with mechanistic models to reduce their computational cost. They are very popular nowadays to help researchers move away from pure trial-and-error experimental approaches, to facilitate property calculations, and to ease the parameterization of mechanistic models[32,33]. For instance, Wang et al. applied different ML algorithms to analyze molecular dynamics simulations[34], whereas Adam et al. combined neural networks with physics-based models to improve the accuracy of lithographic process modeling[35]. From an industrial perspective, Maleh et al. studied the use of ML techniques for Internet of Things (IoT) intrusion detection in aerospace cyber-physical systems[36], and Tuncali et al. evaluated cyber-physical systems with ML in the context of autonomous driving systems[37]. In contrast to usual data analysis techniques for multivariate data sets, Functional Data Analysis (FDA), which is typically used to process time series, can be also connected with ML techniques for the forecasting or prognostic of such time series[38]. Therefore, this coupling can constitute a straightforward solution to deal with time-dependent data from mechanistic models in order to contribute in their computational cost reduction[39].

In this work, we tackled the issue of computational cost reduction of 3D-resolved mechanistic models for electrode slurry rheology simulation with molecular dynamics through the use of a functional data-driven framework for fast predictions of their simulation results. Briefly, this framework bases its operation on only executing the first numerical steps (i.e. time frames) of the mechanistic simulation to retrieve early numerical values, and then bypassing the full simulation process by predicting its final results without the need to run it until the end. More precisely, the aforementioned framework proposes first a screening step, whose main goal is to identify running simulations that will end with a result in a range of interest for our manufacturing modeling, e.g. expected to provide results comparable with the experimental data, in order to validate the CGMD mechanistic model of the electrode slurry. Second, it proposes a forecasting step to quickly predict the NEMD results, considering only the previously filtered simulations within the range of interest. Both steps couple two algorithms: one based on Functional Principal Component Analysis (FPCA) achieving compression of the time series in a low dimensional space (i.e. dimensionality reduction), and another one based on K-Nearest-Neighbors (KNN) performing the predictive task. We applied this framework for electrode slurry modeling based on NEMD after accumulating simulations over three different LIB electrode AM chemistries, making the framework extensive to different materials[40]. We tracked the evolution of the calculated viscosity values (η) along the simulation process to determine the *shear-viscosity* curve ($\dot{\gamma}$-η curve). It is calculated point by point, i.e. a shear rate is applied through the deformation of the simulated slurry box to define the time series. Despite the particular illustration here for the case of LIB electrode slurries, the proposed framework can be also applied to other fields where mechanistic models are employed to generate time series data providing a significant computational cost reduction, but also making feasible a faster optimization process.

RESULTS AND DISCUSSION

NEMD calculations of electrode slurry viscosities generate discretized values at each numerical step (time frames) along the calculation process. Those steps are independent of the allocated resources used and produce a list of numerical values to constitute a viscosity curve as a function of time. Such a curve (written here $X(t)$) is the descriptor of one simulation, and the appropriate data treatment of this curve requires fixing the number of time frames for the definition of the corresponding functional space I[41]. The functional form of a viscosity vs. time curve is written

$$\mathbf{X} = \{X(t), t \in I\} \quad I \subseteq \mathbb{R}^+ \tag{1}$$

FDA properly carries out those types of variables by defining highly regular data over the space I through smoothing techniques. FDA has shown to be a very relevant approach in numerous contexts, for instance in clinical studies[42], sports performance analysis[43], or materials discovery[44,45]. Within this study, we applied a FPCA on time series data as a tool to achieve their compression (dimensionality reduction), making possible to recover new variables that will be employed as input values of ML models for further predictive tasks. In particular, FPCA obtains high variability on a meaningful low dimensional representation of time series[46,47]. Such data processing is already well known within the usual principal component analysis (PCA)[48], but here it is extended to the functional form of time series.

FPCA requires a smoothing technique to reconstruct a set of discrete values from time series (equation (1)), over the functional

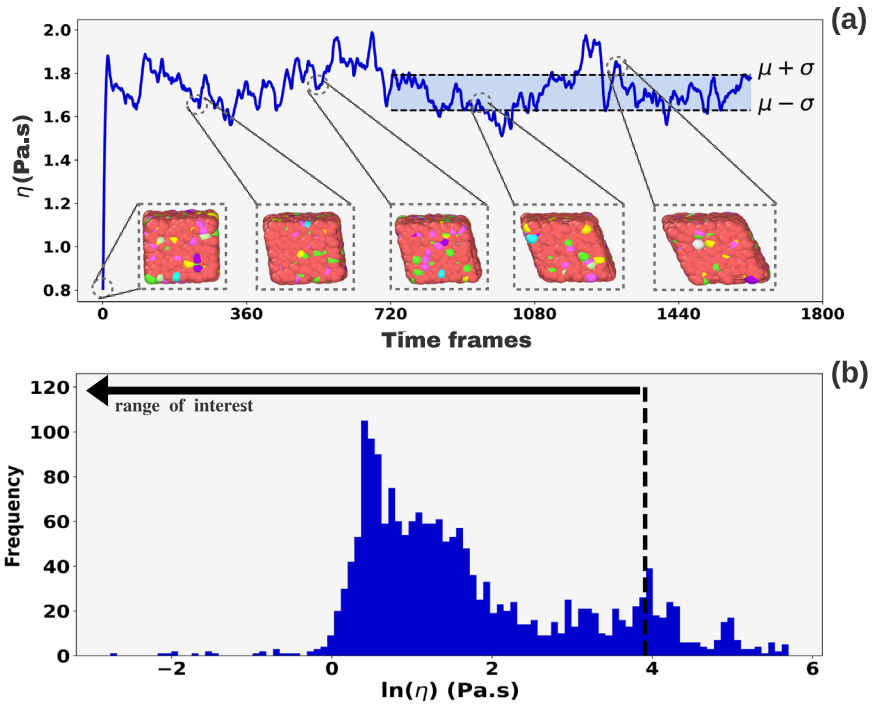

Fig. 2 Data treatment of the NEMD simulation for the assessment of the simulation results. a The slurry microstructure evolution and associated viscosity (η) during the NEMD simulation were handled to assess the slurry viscosity at a given shear rate (γ). The last 1000 viscosity values from a single simulation are used as a vector to calculate an average value (μ) and a standard deviation (σ). These latter values enable to color-code the slurry viscosity evolution within a box, where the two limits are the average plus and minus the standard deviation ($\mu \pm \sigma$). **b** Empirical natural log distribution of the average slurry viscosity value (η) associated with the mechanistic simulations carried out in this study. The black arrow illustrates the range of interest for the simulated viscosity values.

space. The latter space is characterized by a set of basis functions, that enables a basis decomposition of time series in a finite-dimensional space[49]. Considering a time series like the one in equation (1), and a set of basis functions $\phi = \{\phi_1, \phi_2, \ldots, \phi_p\}$ all defined on I (p ≥ 1), $X(t)$ is given by

$$\mathbf{X} = X(t) = \sum_{i=1}^{p} c_i \times \phi_i(t) \quad (\mathbf{c_i}, t) \in \mathbb{R} \times I \tag{2}$$

where the set $c = \{c_1, c_2, \ldots, c_p\}$ is the vector of basis coefficients, which describes **X** as a finite number of values.

The basis family most commonly used to perform this type of decomposition is the *B-Spline* family[50]. The latter consists on interpolating time series with a family of polynomial functions (basis functions) which are sufficiently differentiable in discrete knots. Considering $I = [l_0, l_1]$ with $l_0 = t_0 < t_1 < \ldots < t_p = l_1$, each polynomial is defined in sub-intervals of I, expressed by $(t_i)_{(i \leq p)}$ values. These polynomials are positive on at most p sub-intervals, and their degree is equal to 3 in this study. This data processing initializes the FPCA which estimates M eigenfunctions $(\psi_i)_{(i \leq M)}$ associated to M eigenvalues $(v_i)_{(i \leq M)}$, and finally M scores $(\rho_{i,j})_{(i \leq M)}$ representing the projection of the *j-st* time series ($X_j(t)$) along the new axis, also called functional principal components. Following the Karhunen–Loève decomposition, $X_j(t)$ is expressed as $X_j(t) = \sum_{m=1}^{M} \rho_{m,j} \times \psi_m$[51].

In particular, M is very limited compared to the initial length of the time series, since the FPCA must keep as much variability (variance) as possible from them in a short set of functional principal components, which illustrates the concept of dimensionality reduction (see the Supplementary Methods for the mathematical explanations of the FPCA). In the end, the resulting M scores are sufficient to describe each time series in a discrete manner and can be used as input values to embed further supervised ML implementations of our framework for predictive tasks. The main advantages concern the possibility to decrease the training time due to a narrowed number of possible input variables combinations, but also to avoid overfitting during the training step[52,53]. The latter-mentioned fitting usually happens when the feature space becomes increasingly sparse with a high number of parameters (inputs), since ML models can be very sensitive to a small set of parameters[54]. Therefore, the application of dimensionality reduction for time series helps, in that sense, to represent a meaningful data processing for defining inputs of ML models. Table 2 in the Methods section illustrates the number of inputs per algorithm. In our framework, we coupled such a FPCA with a supervised ML algorithm to perform two different predictive tasks, being the two pillars of our study.

Simulations in the range of interest

To speed up the validation of the electrode slurry simulation, we are interested on predicting in advance the range in which the values of the viscosity NEMD calculations will fall in the end, allowing us to keep only the X(t) that will likely reach values close to the experimental ones. Indeed, it is not useful to keep running electrode slurry simulations which offer viscosity values that are likely to be far away from the experimental values. More precisely, in our viscosity tests we applied a shear rate (deformation of the slurry microstructure illustrated in Figure 2a) of 0.1–1000 s^{-1} corresponding to viscosity values between 0.1 and 50 Pa s. It is therefore reasonable to model the viscosity in this range of values, since these are the viscosity values that make the electrode slurry dense. As a result, we consider predicting whether a running simulation will end up with a result in this range of values. Otherwise, it means that this simulation will provide viscosity values that are too far from the expected experimental values and that it can be stopped in order to free up the hardware used for

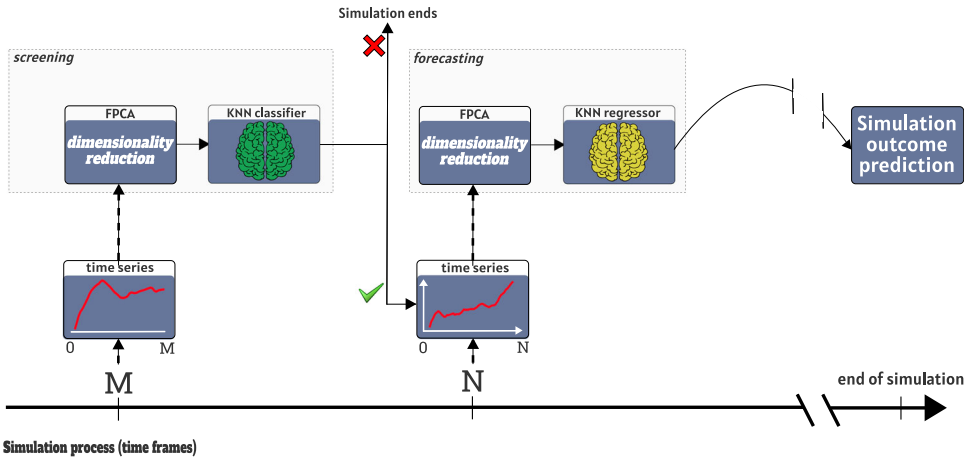

Fig. 3 Schematic representation of the functional framework with the different steps to fast forecast the mechanistic simulation results. *M* and *N* are the two quantities considered of numerical values already generated by the simulation to assess the time series for the screening and the forecasting steps. Such a framework illustrates how a running simulation is handled during its processing taking only into account the early numerical steps, to then predict its final results and decrease the overall computational cost.

the computations and to run another simulation associated with other conditions.

The simulation result is calculated on the final behavior of the simulation as Figure 2a displays. In fact, we used the average of the numerical values resulting from the last 1000 time frames to assess the result of a simulation (i.e. μ). We labeled each NEMD simulation by a binary value representing if its result is falling into the range of values [0.1; 50] Pa.s. Then, we determined the first pillar (called supervised classification in the ML terminology) of our framework, achieving the screening of mechanistic simulations in the very first numerical steps, by predicting if a running simulation will be ending with a result in the range of interest defined above. The prediction provided concludes on the interest of a running mechanistic simulation to let it continue for forecasting its viscosity result. This represents an automatized *go* or *no go* decision in our framework (Figure 3) that allow us to only focus on running simulations that are in a range of interest due to similarities with experiments through the analysis of how the simulation behaves within the early numerical values.

K-Nearest-Neighbors

This screening step (classification step) within the aforementioned framework is executed using a KNN classification[55]. One basis of the algorithm is the memorization of distances between data from the features space. The latter, which is defined by the functional principal components from the FPCA, enhances calculations of sparse distances between time series (inputs) to make a meaningful choice for our study[56]. The KNN algorithm relies on the choice of k neighbors and a distance metric to evaluate pairwise distances between input data, to then attribute a prediction (output) for unseen input data based on the outputs of its k nearest neighbors[57]. Moreover, the right choice of k is crucial to take into account a suitable number of neighbors when predicting the output. This affects the predictive capabilities of the KNN. In this direction, the algorithm is straightforward and can be summarized as follows (see Supplementary Figure 2 for the graphical interpretation of the KNN algorithm):

i. Choose the number of neighbors k;
ii. Calculate the distance between data considering a specific metric;
iii. Get the k nearest neighbors for another data we want to predict;
iv. Assign the associated output by a majority vote (classification), or by an average value (regression), for the outputs of its k nearest neighbors.

From the perspective of the framework, the chosen number of time frames to determine the time series in the screening step must be low, since the actual efficiency achieved when making the prediction within the early numerical steps is mainly determined by this parameter. Moreover, this enables us to stop quickly a running simulation if the latter does not fall within the range of interest for our slurry modeling. This is particularly meaningful in our accelerated computational methodology for freeing up computational resources and then launching another simulation with other operating conditions. This is discussed in more detail below by comparing the predictive capabilities of the KNN classification as a function of the number of time frames for the time series definition.

Fast forecasting from mechanistic simulations

The second pillar of the aforesaid framework concerns the fast prediction of the results, only for mechanistic simulations that have been previously considered in the range of interest, as Figure 3 displays. In this second stage, the results represent now the complete information we can extract from the last numerical steps of the simulation. As Figure 2a shows, it is not only the average value (μ) but also the standard deviation (σ) from the vector formed by the numerical values of slurry viscosities within the last 1000 time frames. This collection of numerical values was empirically set to deal with the variability of the viscosity behavior along the simulation process, based on the available mechanistic simulations in the dataset (Figure 4 encompasses various viscosity behaviors to emphasize the application of our functional framework for the data processing of viscosity curves for its forecasting purpose). These results not only provide the final average viscosity of the associated simulation but also the corresponding standard deviation of the final viscosity, which describes how the last 1000 viscosity values behave around μ. To achieve this forecasting step, we also couple a FPCA with a KNN algorithm, but in the context of a supervised regression for the latter algorithm considering in this case two outputs, i.e. $\mu-\sigma$, and $\mu+\sigma$.

In addition to the previous time series assessment in the screening step, we used another number of time frames to recover additional numerical values of the ongoing simulations to define the time series (Figure 3) for the dimensionality reduction

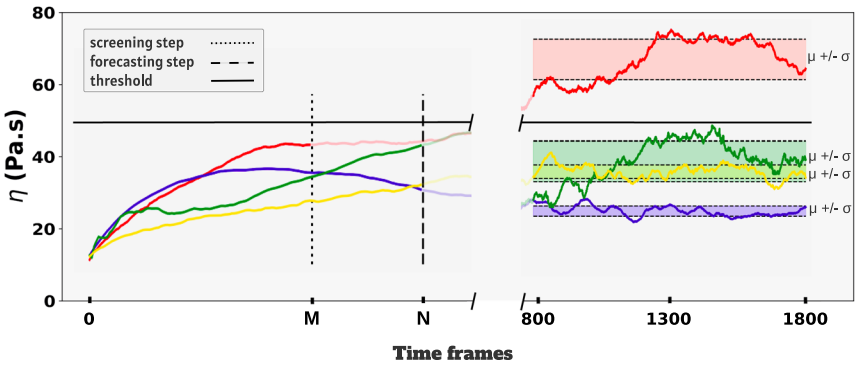

Fig. 4 Graphical representation of a few examples of mechanistic simulations with various slurry viscosity evolutions along with the time frames. We highlight the early numerical values to represent the viscosity curve taken into account in the screening, and forecasting steps if applicable, as well as the last 1000 numerical values serving for the calculation of associated simulation results. As it can be noticed, the numerical values in between are removed for the schematic representation of the framework's purpose. See Supplementary Figure 3 for the full-length simulations corresponding to these examples. *M* and *N* still represent the time frames selected to define time series for the screening and forecasting step. The threshold represents the upper limit for our range of interest.

here in the forecasting step. In practice, a simulation is executed until a convergence criteria is satisfied and a stable value is reached to define its final result (Figure 4). This suggests that by recovering the numerical values within a large number of time frames, we may expect the KNN regression to provide predictions that are very close to the simulation result. Moreover, this action does not reduce the computational cost since the simulation runs for a large number of time frames. As a consequence, the forecasting step balances the number of time frames to use for the definition of the time series, and that will define the corresponding computational cost to fast predict the simulation result, with the level of predictive capabilities for the KNN. We compared these predictive capabilities of the KNN regression (as for the classification) for different numbers of time frames to find the best compromise. The results are discussed below.

Datasets

The time series were extracted from NEMD simulations to capture the rheological behavior of LIB slurries (viscosity vs. shear rate). Simulations were launched using the LAMMPS software[10]. In total, we executed 2172 simulations coming from the modeling of three different slurries to build a complete data set: 1773 from Nickel-Manganese-Cobalt (NMC), 183 from graphite, and 216 from Lithium-Iron-Phosphate (LFP) slurries. We noticed that the type of chemistry does not determine an input value for the KNN algorithm, since only the corresponding rheological behavior was used for our data processing. The LAMMPS code was executed using the MatriCS HPC platform from the Université de Picardie Jules Verne (Amiens, France)[58]. In total, the number of time frames to reach stable viscosity values (η) was between 1450 and 1800, with an average number equal to 1734. In the screening step, all the executed simulations were used as a functional dataset to train and test our KNN classification. The training dataset contained 80% of simulations randomly selected from the functional dataset, and the testing dataset contained the remaining 20%. In the forecasting step, only simulations labeled in the range of interest were used to form another functional dataset to train and test our KNN regression, whose size is equal to 1927 simulations.

Time frames selection

We chose the number of time frames to set each time series according to what is presented in Figure 5, where the trends for the predictive capabilities, i.e. validation metrics, as a function of the number of time frames to set time series for the screening and forecasting step are displayed. Figure 5c reflects a straightforward graphical representation of the overall compromise between the maximization of the validation metrics and the maximization of the computational efficiency (inversely proportional to the computational cost), illustrating a desirable *best compromise* found in between. In this context, the higher the time required to predict the simulation results, the lower the computational efficiency achieved. For each selected number of time frames, we calculated a numerical mean of the validation metrics by generating a random training/testing dataset 20 times. It is particularly important to avoid a biased calculation of the validation metrics by only considering one single random split of the data into the training and testing datasets. The selected validation metrics and their optimization through hyperparameter tuning are detailed in the Methods section below.

We retained a number of time frames equal to 100 for the screening step, whereas the number for the forecasting step was equal to 150. As expected, Figure 5a and b show increasing trends in the validation metrics when incrementing the number of time frames, while they start to stagnate from those two chosen numbers. In the end, the corresponding validation metrics were on average equal to 0.84 and 0.90 for the *Recall* and $F1_{score}$ (KNN classification), and equal to 0.97 and 0.96 for the R^2_{score} of both outputs (KNN regression).

Computational cost decrease

In terms of the computational cost reduction, the framework enables us to use time series defined over 100 (screening step) and 150 (forecasting step) time frames to accurately predict the different results from a running simulation. Considering the average number of time frames to execute a mechanistic simulation in the common slurry modeling approach detailed above (i.e. 1734 time frames), we have successfully developed an efficient framework that is able to determine the slurry viscosity 11 times faster. Furthermore, the same framework determines if the final result of a running simulation will fall or not into the expected range of experimental values 17 times faster. These results illustrate a drastic reduction of the time required to recover the simulation outputs, and also to provide quickly some information on how a given simulation behaves. In addition, it is essential to discard running simulations that are not in the range of interest, corresponding to viscosity values far too high to be representative of any of our experimental conditions[5]. Moreover, the fast prediction of simulation results enables the allocation of free available resources on an HPC platform for another simulation to fast study other operating conditions. In the battery manufacturing context, such an advantage facilitates to seek the best parameter set that

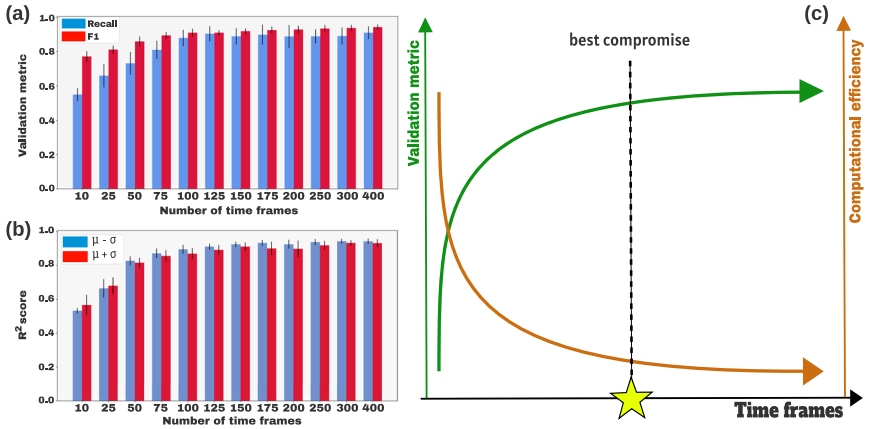

Fig. 5 Time frame selection based on the validation metrics from the training of different KNN models. a Evolution of the average *Recall* and $F1_{score}$ as a function of the number of time frames. The latter defines the time series needed for the FPCA within the screening step of the framework. The black slight bar represents the standard deviation (SD) around the mean value of the validation metric. **b** Evolution of the average R^2_{score} as a function of the number of time frames for each KNN output. The latter defines the time series needed for the FPCA within the forecasting step of the framework. **c** Graphical representation of the best compromise between the validation metrics and the corresponding computational efficiency to calculate the simulation results for the time frame selection. The green trend illustrates the results from **a** and **b** and the orange one is a simplification of the evolution of the computational efficiency function of time frames to clarify the overall compromise. The expected best compromise is illustrated where both trends plateau.

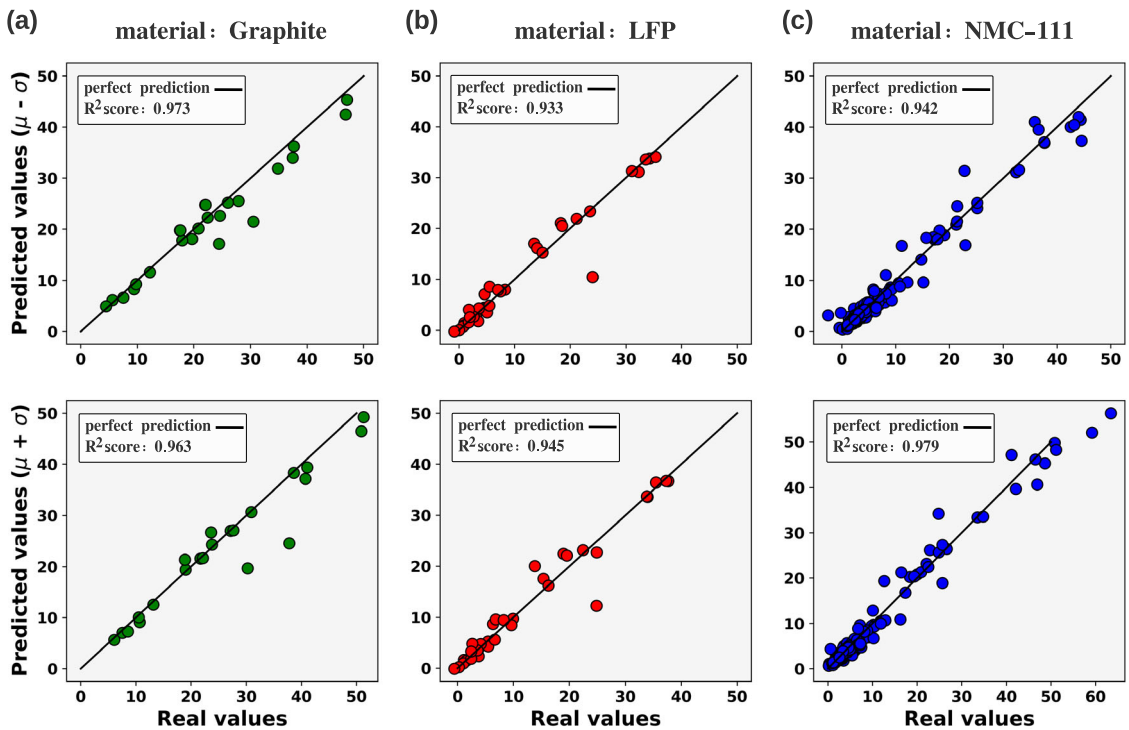

Fig. 6 Regression plots to evaluate the predictive capabilities of the KNN regression for the fitting of $\mu \pm \sigma$. a Comparison of the predicted values against the real values, for mechanistic simulations associated with the Graphite within the testing data set. **b** Comparison of the predicted values against the real values, for mechanistic simulations associated with the LFP within the testing data set. **c** Comparison of the predicted values against the real values, for mechanistic simulations associated with the NMC-111 within the testing data set.

maximizes a given observable among extensive possibilities, triggering further and faster optimization processes of mechanistic model parameters.

Chemistry neutral approach

According to the R^2_{score} corresponding to the two KNN's outputs ($\mu \pm \sigma$) that were calculated with the testing dataset, we can observe that our framework is able to forecast simulation results in a highly reliable manner. Figure 6 displays the *regression plots* related to the retained KNN regression. We compared the predicted values against their corresponding real full-length simulation results by separating the testing dataset for each material, despite the fact that the training process did not take into account the chemistry as an input value. This graphical representation reflects how accurate is the framework

to make predictions regardless of the specific chemistry considered during the mechanistic simulation. Therefore, we calculated the R^2_{score} over each material and concluded, as it can be also seen in the same Figure 6, that the framework provides overall high validation metrics without differences in the results. Besides these global predictive capabilities, our framework has the specificity of being chemistry neutral, enlarging the practical application of our tool for the slurry modeling to any of the numerous chemistry used in the battery field during this first step of the battery manufacturing process.

In this study, we presented a functional data-driven framework for fast predictions of mechanistic simulation results, from the very first numerical iterations without the need to run the simulation until the end. We demonstrated this data-driven approach for the case of viscosity calculations versus applied shear rate for LIB electrode slurries, with different AM chemistries. Such calculations are supported on NEMD and are usually computationally expensive. The aforementioned framework quickly determined slurry viscosities within the forecasting step, leading to a computational cost reduction by a factor 11. Besides, such a framework also enabled us to predict if a simulation will end within the range of interest for experimental comparisons 17 times faster within the screening step. This early determination is enhanced by the inclusion of mechanistic simulations coming from different materials, making it possible to apply the framework for chemistries that are widely used in the battery field. In terms of predictive capabilities, the KNN classification reached an $F1_{score}$ equals to 0.90, whereas the KNN regression achieved overall R^2_{score} equal to 0.97 and 0.96 for both KNNs outputs. Moreover, we obtained the same range of values when computing the validation metrics for each specific material. These validation metrics show the high predictive capabilities of the framework regardless of the chemistry associated with a mechanistic simulation. In particular, the results presented in this work demonstrate that our framework is able to treat mechanistic simulations as time series, for the application of FDA techniques to embed supervised learnings performing predictive tasks. In this way, it serves as a tool to support the training process of those ML algorithms aimed at analyzing the rheology of the electrode slurries. As a perspective, it is important to highlight that the proposed framework can be transferred to any other field where mechanistic models generate time series, to then benefit from a drastic decrease in the corresponding computational costs. This is the case on a plethora of physical scales, e.g. materials properties from their electronic structure[59] or the dynamics of colliding galaxies derived from gravitational forces[60] and gas hydrodynamics[61]. In the context of battery modeling, other steps of the manufacturing chain can be also optimized with such an accelerated computational methodology, for instance the DEM modeling of the calendering process which tracks electrode properties along time (e.g. porosity).

METHODS

Validation metrics

We evaluated the performance of both the KNN classification and the regression by using two different metrics on the testing data sets. The $F1_{score}$[62] and, especially, the corresponding Recall score were used to validate the KNN classification, whereas we assessed the goodness of the fitting for the KNN regression with the R^2_{score}[63]. The first mentioned metric corresponds to the harmonic mean of the *Precision* and *Recall* (equation (3)) and from its optimization a binary classifier should reduce the number of incorrect predictions and simultaneously increase the number of correct ones. In our context, this is especially relevant due to the large number of mechanistic simulations with a final result lower than the threshold of 50 Pa.s, compared to the number of mechanistic simulations with a final result greater than 50 Pa.s. within the dataset. As an example, we considered minimizing the error from the KNN classification to predict a running simulation, although its real full-length result is far too high to be representative of the expected experimental conditions. This is reported in Table 1 with the confusion matrix corresponding to the KNN classification instances.

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

where the Recall allows minimizing *FN*, as expressed above[64].

The second mentioned metric (R^2_{score}) was computed to assess how close are the predicted simulation results from their real full-length simulation results. The higher the score, the lower the discrepancy between the prediction and the real value. Such a score is calculated as follows

$$R^2_{score} = 1 - \sum_{i=1}^{n} (y_i - \tilde{y_i})^2 / \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{4}$$

where y_i, $\tilde{y_i}$, and $\bar{y}$ are the predicted values by the deterministic learning, the real value from the data set, and the average of the real values respectively.

Computational details

The functional framework applies an FPCA on time series for the screening and forecasting step, using a B-spline decomposition with a set of knots whose size is equal to 60% of the time series length. FPCAs provided a set of functional principal components corresponding to 99% of the initial variability (variance) from the initial data set. Table 2 summarizes the different number of input values regarding each algorithm within the framework. This illustrates how the latter narrows the overall information from the time series, reducing the number of input values which is beneficial for the KNNs training process.

Hyperparameters tuning

All validation metrics within our study were obtained by tuning the hyperparameters of both KNN classification and regression[65]. In practice, the number of nearest neighbors (k) is difficult to assess while initializing a random value. To deal with this, we used cross-validation (CV) through the *GridsearchCV* function[66] available within the common *Scikit-Learn* library of Python. We empirically set the number of CV splits at five to not increase the training time. Table 3 reflects the results obtained for the hyperparameters tuning.

Table 1. Summary of the possible instances, i.e., predictions, in the case of the KNN classification, for the screening of mechanistic simulations.

KNN classification' instance	Interpretation	Confusion matrix
True positive (TP)	Irrelevant simulations predicted as irrelevant	10.5%
True negative (TN)	Relevant simulations predicted as relevant	85.5%
False positive (FP)	Relevant simulations predicted as irrelevant	2%
False negative (FN)	Irrelevant simulations predicted as relevant	2%

For the sake of simplicity, we replaced the meaning of a simulation to end within the range of interest by relevant, and its opposite case by irrelevant.

Table 2. Summary of the number of values taking action of the screening step (classification) and the forecasting step (regression) in the functional framework.

Parameters	Screening	Forecasting
Initial length of time series	100	150
Number of knots in the basis decomposition	60	90
Number of functional principal components	10	15
Number of input values for the KNN	10	15

The initial length of the time series can be compared with the number of input values of both KNNs, illustrating the dimensionality reduction from the functional data analysis.

Table 3. Hyperparameters tuning from the cross-validation, for the KNN classification and regression.

Hyperparameter	Screening	Forecasting
Number of nearest neighbors	$k = 28$	$k = 7$
Distance metric	Manhattan	Minkowski
Power parameter	1	2

Others KNN hyperparameters are set with initial values proposed by the function.

DATA AVAILABILITY

This study did not generate new unique materials. Data and codes are available upon reasonable request. They will be made available in ARTISTIC-ERC/FunctionalSimulations (official ARTISTIC project GitHub page) and MarcDuquesnoy/FunctionalSimulations.

Received: 7 January 2022; Accepted: 30 May 2022;
Published online: 22 July 2022

REFERENCES

1. Dean, T., Allen, J. & Aloimonos, Y. Artificial intelligence: theory and practice (Benjamin-Cummings Publishing Co., Inc., 1995).
2. Patterson, D. Introduction to artificial intelligence and expert systems (Prentice-Hall, Inc., 1990).
3. Kohl, M. et al. Physical model for the spectroscopic analysis of cortical intrinsic optical signals. *Phys. Med. Biol.* **45**, 3749 (2000).
4. Ngandjong, A. C. et al. Investigating electrode calendering and its impact on electrochemical performance by means of a new discrete element method model: towards a digital twin of li-ion battery manufacturing. *J. Power Sources* **485**, 229320 (2021).
5. Lombardo, T. et al. Accelerated optimization methods for force-field parametrization in battery electrode manufacturing modeling. *Batter. Supercaps* **3**, 721–730 (2020).
6. Maingi, V. et al. Stability and dynamics of membrane-spanning dna nanopores. *Nat. Commun.* **8**, 1–12 (2017).
7. Warfield, S. K. et al. Real-time biomechanical simulation of volumetric brain deformation for image guided neurosurgery. 23–23 (SC '00: Proceedings of the 2000 ACM/IEEE Conference on Supercomputing, 2000).
8. Souza, P. C. T. et al. Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nat. Methods* **18**, 382–388 (2021).
9. Sloane, C. S. & Wolff, G. T. Prediction of ambient light scattering using a physical model responsive to relative humidity: validation with measurements from detroit. *Atmos. Environ.* **19**, 669–680 (1967).
10. FrantzDale, B., Plimpton, S. J. & Shephard, M. S. Software components for parallel multiscale simulation: an example with lammps. *Eng. Comput.* **26**, 205–211 (2010).
11. MathWorks. Simulink. https://fr.mathworks.com/products/simulink.html. Accessed on 07.18.2021.
12. Cloud, G. Vertex AI. https://cloud.google.com/ai-platform. Accessed on 07.18.2021.
13. Microsoft, A. Azure Machine Learning. https://azure.microsoft.com/fr-fr/services/machine-learning/. Accessed on 07.18.2021.
14. Slezak, L. Annual progress report 2009, office of freedom car and vehicle technologies. US Department of Energy, Washington (2009).
15. Duquesnoy, M. et al. Data-driven assessment of electrode calendering process by combining experimental results, in silico mesostructures generation and machine learning. *J. Power Sources* **480**, 229103 (2020).
16. Thomitzek, M. et al. Simulating process-product interdependencies in battery production systems. *Procedia CIRP* **72**, 346–351 (2018).
17. Li, J., Fleetwood, J., Hawley, W. B. & Kays, W. From materials to cell: state-of-the-art and prospective technologies for lithium-ion battery electrode processing. *Chem. Rev.* **122**, 903–956 (2021).
18. Cunha, R. P., Lombardo, T., Primo, E. N. & Franco, A. A. Artificial intelligence investigation of NMC-cathode manufacturing parameters interdependencies. *Batter. Supercaps* **3**, 60–67 (2020).
19. Turetskyy, A. et al. Toward data-driven applications in lithium-ion battery cell manufacturing. *Energy Technol.* **8**, 1900136 (2020).
20. Turetskyy, A., Wessel, J., Herrmann, C. & Thiede, S. Battery production design using multi-output machine learning models. *Energy Storage Mater.* **38**, 93–112 (2021).
21. Fichtner, M. et al. Rechargeable Batteries of the Future—The State of the Art from a ATTERY 2030+ Perspective. *Adv. Energy Mater.* https://doi.org/10.1002/aenm.202102904 (2021).
22. Shodiev, A. et al. 4D-resolved physical model for electrochemical impedance spectroscopy of Li(Ni1-x-yMnxCoy)O2-based cathodes in symmetric cells: consequences in tortuosity calculations. *J. Power Sources* **454**, 227871 (2020).
23. Chouchane, M., Rucci, A., Lombardo, T., Ngandjong, A. C. & Franco, A. A. Lithium ion battery electrodes predicted from manufacturing simulations: assessing the impact of the carbon-binder spatial location on the electrochemical performance. *J. Power Sources* **444**, 227285 (2019).
24. Chouchane, M., Primo, E. N. & Franco, A. A. Mesoscale effects in the extraction of the solid-state lithium diffusion coefficient values of battery active materials: physical insights from 3D modeling. *J. Phys. Chem. Lett.* **11**, 2775–2780 (2020).
25. Chouchane, M., Arcelus, O. & Franco, A. A. Heterogeneous solid-electrolyte interphase in graphite electrodes assessed by 4D-resolved computational simulations. *Batter. Supercaps* **4**, 1457–1463 (2021).
26. Lombardo, T. et al. The ARTISTIC online calculator: exploring the impact of li-ion battery electrode manufacturing parameters interactively through your browser. *Batter. Supercaps* **5**, 1–9 (2022).
27. Toffolo, A., Masi, M. & Lazzaretto, A. Low computational cost cfd analysis of thermoacoustic oscillations. *App. Therm. Eng.* **30**, 544–552 (2010).
28. Bradley, E. G. & Kendall, B. A review of computer simulations in teacher education. *J. Educ. Technol. Syst.* **43**, 3–12 (2014).
29. Li, J. & Geng, S. Industrial clusters, shared resources and firm performance. *Entrep. Reg. Dev.* **24**, 357–381 (2012).
30. Mochalov, V. P., Linets, G. I. & Palkanov, I. S. Methods and models of resource allocation in load balancing clusters. In Computer Science Online Conference. vol. 230, 552–563 (Springer International Publishing, Cham, 2021).
31. Elisa, G. & Vittorio, V. Compact physical model for simulation of thermal networks. *Energy* **175**, 998–1008 (2019).
32. Scherer, C., Scheid, R., Andrienko, D. & Bereau, T. Kernel-based machine learning for efficient simulations of molecular liquids. *J. Chem. Theory Comput.* **16**, 3194–3204 (2020).
33. Deosarkar, P. M. & Sathe, S. V. Predicting effective viscosity of magnetite ore slurries by using artificial neural network. *Powder Technol.* **219**, 264–270 (2012).
34. Wang, Y., Marcelo, J., Ribeiro, L. & Tiwary, P. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Batter. Supercaps* **61**, 139–145 (2020).
35. Adam, K. et al. Using machine learning in the physical modeling of lithographic processes. In *Design-Process-Technology Co-optimization for Manufacturability* XIII, vol. 10962, 86–93 (SPIE, 2019).
36. Maleh, Y. Machine learning techniques for iot intrusions detection in aerospace cyber-physical systems. In *Machine Learning and Data Mining in Aerospace Technology*, 205–232 (Springer, 2020).
37. Tuncali, C. E., Fainekos, G., Ito, H. & Kapinski, J. Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 1555–1562 (IEEE, 2018).
38. Guo, J. & Li, Z. Prognostics of lithium ion battery using functional principal component analysis. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 14–17 (IEEE, 2017).
39. Pal, M. On application of machine learning method for history matching and forecasting of times series data from hydrocarbon recovery process using water flooding. *Pet. Sci. Technol.* **39**, 519–549 (2021).
40. Giménez, C. S. et al. Numerical simulation of the behavior of lithium-ion battery electrodes during the calendaring process via the discrete element method. *Powder Technol.* **349**, 1–11 (2019).

41. Petersen, A. & Muller, H.-G. Functional data analysis for density functions by transformation to a hilbert space. *Ann. Statis.* **44**, 183–218 (2016).
42. Donoghue, O., Harrison, A., Coffey, N. & Hayes, K. Functional data analysis of running kinematics in chronic achilles tendon injury. *Med. Sci. Sports Exerc.* **40**, 1323–1335 (2008).
43. Pataky, T. C. One-dimensional statistical parametric mapping in python. *Comput. Methods Biomech. Biomed. Engin.* **15**, 295–301 (2012).
44. Cheng, Y., Lu, C., Li, T. & Tao, L. Residual lifetime prediction for lithium-ion battery based on functional principal component analysis and bayesian approach. *Energy* **90**, 1983–1993 (2015).
45. Guo, J. & Li, Z. Prognostics of lithium ion battery using functional principal component analysis. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 14–17 (IEEE, 2017).
46. Nicol, F. Functional principal component analysis of aircraft trajectories. In *ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management* (2013).
47. Happ, C. & Greven, S. Multivariate functional principal component analysis for data observed on different (dimensional) domains. *J. Am. Stat. Assoc.* **113**, 649–659 (2016).
48. Shlens, J. A tutorial on principal component analysis. arXiv preprint https://arxiv.org/abs/1404.1100 (2014).
49. Chagny, G. Statistique pour données fonctionnelles. http://gchagny.perso.math.cnrs.fr/CoursFDA.pdf. Accessed on 07.18.2021.
50. Basdevant, C. On calculating with b-splines. *J. Approx. Theory* **6**, 50–62 (1972).
51. Panaretos, V. M. & Tavakoli, S. Cramer-karhunen-loève representation and harmonic principal component analysis of functional time series. *Stoch. Process. Their Appl.* **123**, 2779–2807 (2013).
52. Paris, G., Robilliard, D. & Fonlupt, C. Exploring overfitting in genetic programming. *In International Conference on Artificial Evolution (Evolution Artificielle)*, 267–277 (Springer, 2003).
53. Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **1168**, 2779–2807 (2019).
54. Matcha, B. B., Chachira, R., Transtrum, M. K. & Sethna, J. P. Parameter space compression underlies emergent theories and predictive models. *Science* **342**, 604–607 (2013).
55. Cover, T. M. Estimation by the nearest neighbor rule. *IEEE Trans. Infor. Theory* **14**, 50–55 (1968).
56. Raschka, S. & Mirjalili, V. Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow 2 (Packt Publishing Ltd, 2019).
57. Cover, T. M. & Hart, P. E. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**, 21–27 (1967).
58. Hager, G. & Wellein, G. Introduction to high performance computing for scientists and engineers (CRC Press, 2010).
59. Chelikowsky, J. R., Saad, Y., Öğüt, S., Vasiliev, I. & Stathopoulos, A. Electronic structure methods for predicting the properties of materials: grids in space. *Phys. Stat. Sol.* **217**, 173–195 (2000).
60. Barnes, J. E. & Hernquist, L. E. Computer models of colliding galaxies. *Phys. Today* **46**, 54–61 (1993).
61. D'Humières, D., Lallemand, P. & Frisch, U. Lattice gas models for 3d hydrodynamics. *EPL* **2**, 291–297 (1986).
62. Popovic, M. chrf: character n-gram f-score for automatic mt evaluation. *Proceedings of the Tenth Workshop on Statistical Machine Translation*. 392–395 (2015).
63. Frost, J. How to interpret r-squared in regression analysis. *Statistics by Jim* (2018).
64. Buckland, M. & Gey, F. The relationship between recall and precision. *J. Am. Soc. Inf. Sci.* **45**, 12–19 (1994).
65. Yang, L. & A. Shami, A. On hyperparameter optimization of machine learning algorithms: theory and practice. *Neurocomputing* **415**, 295–316 (2020).
66. Granholm, V., Noble, W. S. & Käll, L. A cross-validation scheme for machine learning algorithms in shotgun proteomics. *BMC Bioinf.* **13**, 1–8 (2012).

ACKNOWLEDGEMENTS

The authors acknowledge the European Union's Horizon 2020 research and innovation program for the funding support through the European Research Council (grant agreement 772873, "ARTISTIC" project: ARTISTIC-ERC. M.D. and A.A.F. acknowledge the ALISTORE European Research Institute for funding support. A.A.F. acknowledges the Institut Universitaire de France for the support. A.A.F. and F.C. acknowledges the European Union's Horizon 2020 research, and innovation program under grant agreement no. 957189 (BIG MAP). We acknowledge Daphne Boursier at LRCS for the English proofreading of the article and useful comments.

AUTHOR CONTRIBUTIONS

A.A.F. and M.D. had the original idea of the concept presented in this paper. M.D. was in charge of the data-driven development, the data analysis, and the writing of the original version of the manuscript. T.L. was in charge of the generation of NEMD simulations for NMC-based and graphite-based slurries and supported the writing and the data analysis. F.C. proofread the manuscript and provided useful comments. J.X. and F.H. were in charge of the generation of NEMD simulations for LFP-based slurries. A.C.N. and H.O. supported the conceptualization of this work. A.A.F. obtained the funding (ERC Consolidator Grant), supervised the project and the preparation of the manuscript, and carried out the revision.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-022-00819-2.

Correspondence and requests for materials should be addressed to Alejandro A. Franco.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.