SCIENTIFIC REPORTS
nature research

OPEN

A machine learning framework for computationally expensive transient models

Prashant Kumar[1,4], Kushal Sinha[2,3]✉, Nandkishor K. Nere[2,3], Yujin Shin[1,5], Raimundo Ho[1], Laurie B. Mlinar[3] & Ahmad Y. Sheikh[1]

Transient simulations of dynamic systems, using physics-based scientific computing tools, are practically limited by availability of computational resources and power. While the promise of machine learning has been explored in a variety of scientific disciplines, its application in creation of a framework for computationally expensive transient models has not been fully explored. Here, we present an ensemble approach where one such computationally expensive tool, discrete element method, is combined with time-series forecasting via auto regressive integrated moving average and machine learning methods to simulate a complex pharmaceutical problem: development of an agitation protocol in an agitated filter dryer to ensure uniform solid bed mixing. This ensemble approach leads to a significant reduction in the computational burden, while retaining model accuracy and performance, practically rendering simulations possible. The developed machine-learning model shows good predictability and agreement with the literature, demonstrating its tremendous potential in scientific computing.

Machine learning has emerged as one of the most promising technologies in the past decade due to its capability to provide valuable insights[1] into vast amounts of data generated during the Internet era. Rapid democratization of machine learning tools has allowed for the successful adoption of the technology in a wide range of fields including robotics, computer vision[2], speech and natural language processing[3], autonomous driving[4], neuroscience, drug-discovery[5] and in fundamental sciences[6]. However, its application to computational sciences, and applied computational physics in general, has been limited. Prior efforts to apply machine learning to computational sciences have primarily focused on steady state problems which are more tractable. However, applications of machine learning to time-variant problems are rare.

Over the past decade, a tremendous growth in computational power, easily accessed through cloud computing platforms, has been observed. Even then, simulations based on first-principles models of natural systems and, in particular, time-variant problems of these systems remain prohibitively expensive for most practical applications. First-principles models refers to the models that are based on the physical laws such as Newton's laws of motion and are not merely data-driven. Many of these models, such as molecular dynamics (MD)[7] used for enhancing understanding of molecular arrangements, computational fluid dynamics (CFD)[8] used for understanding flow patterns for both gas and liquid phase, density functional theory (DFT)[9] used for understanding electronic (or nuclear) structure, discrete element methods (DEM)[10] used for understanding motion of particulate systems and, last but not the least, finite element method (FEM)[11] used to measure the structural strength of materials, have immense potential to accelerate research and ultimately change the world around us. Advances in the field of ML and artificial intelligence combined with its rapid democratization, increasing adoption in adjacent fields, and ultimately fueled by the rapid growth of computational power in the form of on-demand cloud computing certainly create an opportunity for ML to be utilized for high-fidelity scientific computing as shown in Fig. 1. This framework allows for the development of more accurate system maps using ML tools which can be utilized for optimization and decision-making.

[1]Solid State Chemistry, Process Research and Development, AbbVie Inc., North Chicago, IL, USA. [2]Cross-functional Modeling Forum, Process Research and Development, AbbVie Inc., North Chicago, IL, USA. [3]Process Engineering, Process Research and Development, AbbVie Inc., North Chicago, IL, USA. [4]Present address: Analysis Group, Boston, MA, USA. [5]Present address: Abbott Laboratories, Abbott Park, Lake Bluff, IL, USA. ✉email: kushal.sinha@abbvie.com

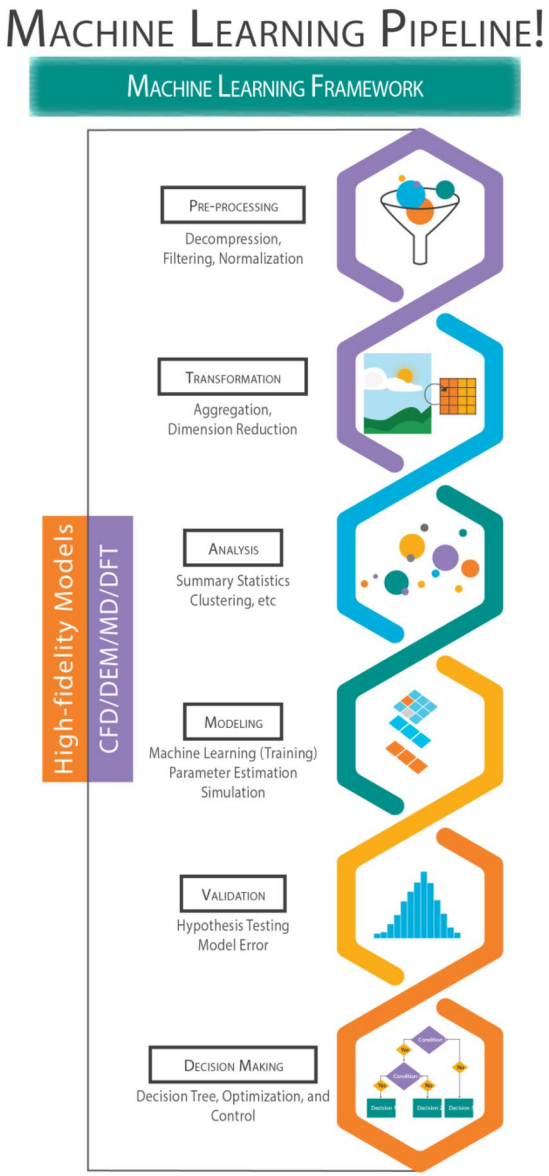

Figure 1. Flowchart of steps involved in applying machine-learning to computationally expensive high-fidelity scientific models. Availability to high-quality data is key to developing a good machine learning predictive model. Identification of meaningful features is paramount to achieving higher model performance. Operations such as data transformation and feature engineering (adding/removing and transforming the available features) enable advanced data inspection, also contributing to better model performance.

Within all of the fields outlined, simplifying assumptions and more computationally affordable coarse-grained representations are required to characterize or predict the overall state of a complex system. However, these simplifying assumptions may ultimately limit the accuracy of the results. Another way of enhancing our understanding with these high-fidelity models is to apply them under idealized conditions or in some regimes of interest. Even then, time-variant simulations of simplified models are highly expensive, as well as unstable, due to restrictions on time steps and other process parameters. The temporal component is either neglected in high-fidelity models or is solved through highly idealized systems of ordinary differential equations which may ignore a lot of relevant details. Most practically relevant transient problems require simulations on the order of hours or days, however stability and computational burden only allow for a few minutes of simulations. Zonal or multi-compartmental modeling[12,13] has been used in some areas such as CFD simulations to overcome the computational cost of transient simulations, however the inherent problem with this approach is the difficulty in defining the zones or compartment that efficiently capture the flow behavior. A great deal of opportunity exists if we can efficiently learn the behavior of the system from a few time-steps (completed in a feasible computational time) and forecast it in time space to remove the need for running computationally expensive simulations until completion.

In light of the above-mentioned challenges, there is an obvious need for a broadly applicable ensemble modeling framework to overcome computational limitations and move towards high-fidelity predictive models that can bridge the gap between coarse-grained systems and real systems in a computationally affordable manner. In order to overcome the challenges of performing transient simulations, we propose the use of a time-series forecasting method known as auto-regressive integrated moving average (ARIMA). ARIMA has been previously used in weather forecasting and stock market prediction[14]; however, its application to first-principle models has not been reported so far to the best of our knowledge. ARIMA can be used to train on data generated from high-fidelity transient simulations and then forecast key relevant physical quantities of interest. As ARIMA is learning from the entire simulation dataset, it has capability to capture start-up transients, local heterogeneities, and temporal evolution of the solution. ARIMA can be an excellent tool to probe the real system under investigation as a function of time. A physical system may have a desired state that can be numerically represented by a time-dependent variable/meta variable reaching a defined value. Hence, ARIMA can be used to forecast the time needed (T_{end}) to reach a desired state, and also the spatial distribution of time-variant physical quantities at that desired state. Taking it a step further, a machine learning predictive model for the time required to reach the desired state (T_{end}) can be built on ARIMA results as a function of multidimensional system parameters. The multidimensional system parameters would be the features of the machine learning trained to predict T_{end}. Machine learning models are quick to probe and preserve the information of high-fidelity models, making it an excellent tool for real-time analysis, optimization, and model-based control of the system of interest.

We selected particulate mixing as our test problem for framework development due to its broad applicability in pharmaceutical, food, and agro-sciences industries. Solid particles mixing is indispensable to achieve desired product quality with respect to content uniformity and reproducible manufacturing across scales in many industrial processes such as drying, blending, and granulation[15,16]. Across the aforementioned applications, understanding of mixing also renders optimal process design and robust scale-up. Controlled mixing can reduce the process cycle time by multiple folds and decrease undesired outcomes, such as particle agglomeration or breakage due to attrition, to ensure optimum product quality. Solid particulate matter and associated processes are complex due to factors such as single particle properties, equipment design, and modes of mixing[16].

An important example of pharmaceutical unit operation which involves particulate mixing is the drying of the active pharmaceutical ingredient (API) in an agitated filter dryer (AFD). Post chemical synthesis and crystallization, the AFD is essential to isolation of potent APIs, where the crystallized API product is separated from the solvents and dried to the desired residual solvent levels. For drying, heat is provided from jacketed vessel walls. Intermittent agitation (or intermittent mixing of the wet cake with the impeller blade) is usually performed to achieve uniform heat transfer across the API bed. The agitation protocol is a key design criterion for this unit operation in which the frequency and duration of agitation and the impeller blade speed can tremendously affect particle properties. An unoptimized agitation protocol will lead to potential agglomeration and/or attrition which would significantly impact the particle size distribution (PSD) achieved at the end of the drying and required for manufacturability and performance of the drug product formulation downstream. Particulate mixing is also relevant to blending and granulation of the drug product formulation and thus impact the dose and content uniformity of each final tablet. Unfortunately, though it is very critical to many pharmaceutical unit operations, particulate mixing is a poorly understood phenomenon.

A first-principles modeling technique, such as DEM, can reveal the underlying mechanistic understanding of particulate mixing. However, like all other high-fidelity scientific computing techniques discussed above, DEM also suffers from the requirement of enormous computing power as practical systems of interest are quite large (i.e. the number of particles are huge and the numerical solutions to compute their motions individually require enormous computing power). For example, a simulation of API particles of 10 μm size in a manufacturing scale filter dryer (0.88 m diameter and fill level of 20 cm) yields a system comprising more than 20 trillion particles which would take around 7,000 CPU core-years to simulate one minute of physical time for particle mixing. Hence, DEM simulations are feasible and limited to systems with a small number of particles or equivalently larger particles for the same fill level. It should be noted that results from a scaled-down model, with small number of particles, cannot not be directly extrapolated for larger systems because of the scale-dependent variability. Computational requirements significantly increase further with cohesive interacting particles or when longer transient simulation times are required. Thus, over the years, a large body of DEM simulations[17–19] performed to understand particulate mixing have limited their investigation to smaller systems.

In this work, we present an ingenious framework for utilizing ARIMA and ML models for computationally expensive transient models (Fig. 2). It should be noted that a very similar route can be taken for other cases. Spatially-averaged segregation index was used in this work to define particle homogeneity in mixing, however a logical extension would be to divide the domain in multiple relevant zones and track desired physical quantities as a function of time in each of these zones, perform ARIMA to predict the time required to reach a desired state and subsequently use ML to map out the entire spatiotemporal evolution of the system. Segregation index, as defined by Eq. 1, represents the extent of mixing of solid particles and is defined based on the spatial position of the particles and the number of contacts between particles of each type.

Results

Segregation index from DEM simulations. DEM simulations of cohesive granular pharmaceutical particles were performed in a manufacturing scale agitated filter dryer. DEM equations are explained in detail in supplementary section S.1. Similar systems can be found in food, agriculture, mining, and chemical industries where particle or powder handling is quite common. In DEM simulations, particle motion is described in a Lagrangian framework wherein equations of motion are solved for each particle or each particle acts as a computational node. At each time step, the forces acting on a particle are computed. A multitude of forces can be acting

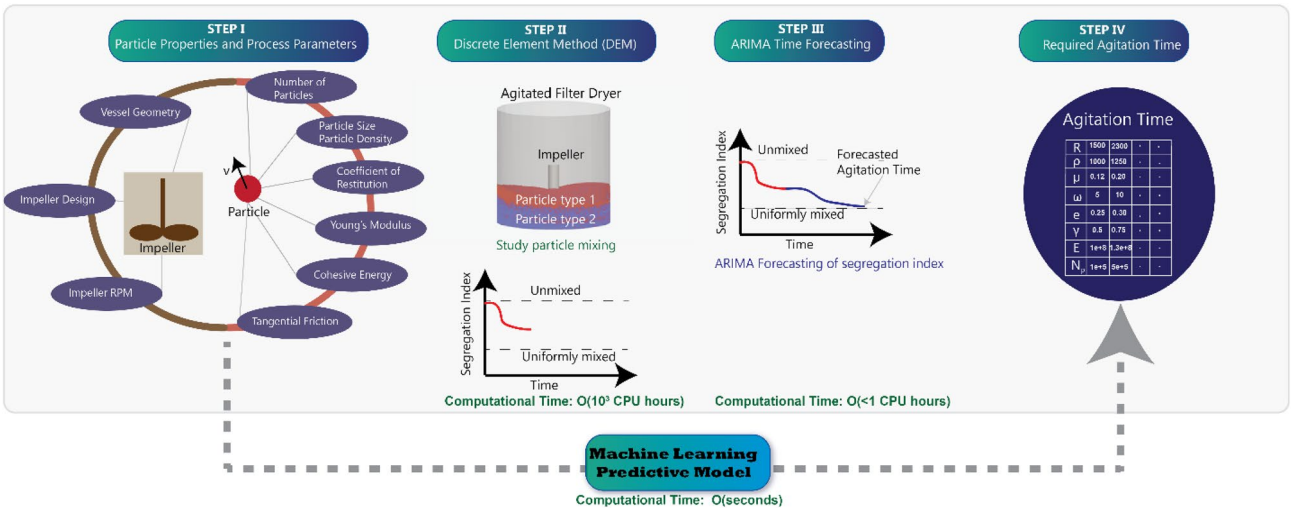

Figure 2. Flowchart of the integrated approach to model solid particle mixing. A machine learning predictive model of solid particle mixing was developed using the integrated approach shown in Fig. 2. DEM simulations (STEP II) should be carried out for some initial time steps to provide the training data for the ARIMA model (STEP III). ARIMA can then be implemented to forecast the mixing behavior and compute the required agitation time. Finally, a machine learning model can be built to predict the agitation time for any set of material properties and process parameters.

at the granular particle scale such as friction, contact plasticity, cohesion, adhesion, liquid bridging, gravity, and electrostatics depending upon the system under study[20–22].

In total, 65 simulations were performed for one minute of physical agitation time by varying the material (particle) properties encompassing a range of particle radius R, particle density ρ, coefficient of restitution e, cohesive energy density $\gamma_{cohesion}$, tangential friction μ_f, Young's modulus E, and process parameters covering a span of number of particles N_p, impeller speed RPM, and cake height h. Impeller RPM represents the rotation speed of the impeller and is reported in revolutions per minute. The typical average time for each of these DEM simulations was over a month. Figure 3 shows the violin plot[23] of the range and frequency of a given parameter in our simulation design space. DEM simulations are initiated with two distinct vertical layers of the particles of types 1 and 2, and the position and velocity of the particles is tracked at all times as shown in Fig. 4a.

The extent of particle mixing is quantified using the Segregation Index parameter, ψ[24], defined in Eq. 1. 'C_{ij}' represents the total number of contacts between particles of type 'i' and 'j' in a given domain. ψ is equal to 1 for uniform random mixing, whereas it is equal to 2 for a completely unmixed scenario as can be seen in Fig. 4b.

$$\psi = \frac{C_{11}}{C_{11} + C_{12}} + \frac{C_{22}}{C_{22} + C_{21}} \tag{1}$$

The asymptotic value of ψ for a system would tend to 1 when approaching random uniform mixing, however the time required depends on a number of factors, which have been investigated in this study. During mixing, spatial arrangement of the particles changes with time resulting in the evolution of the segregation index. At any time, the extent of mixing of particles will be different in different regions of the domain indicating a spatial distribution of ψ as shown in Fig. 4c. The spatial distribution can be attributed to the increase in the linear velocity of the particles along the radial direction resulting in differences in particle collision frequency. Whereas in Fig. 4d, it can be seen that bulk averaged ψ decreases with time during mixing for different impeller angular velocity. With more impeller revolutions, the bed becomes more well-mixed resulting in a drop in the bulk averaged value of ψ. It is clear that longer mixing times would be required at lower RPM as mixing is driven by the number of revolutions. Higher RPM would result in a greater number of revolutions per minute. Even though the bulk averaged ψ approaches to 1, there may be regions closer to the impeller's axis of rotation (regions R1 and R2 in Fig. 4c) where more revolutions would be required for uniform mixing. The caveat with a longer agitation period is that it can affect particle size distribution because of particle attrition[25] and agglomeration[22]. Particle agglomeration and attrition are the key challenges that govern decisions or design of an optimized agitation protocol and need to be prevented to ensure product quality. It is, therefore, crucial to know the approximate agitation time required for uniform mixing for drying effectiveness but avoiding over-mixing which can lead to particle agglomeration and attrition.

Time forecasting of segregation index. Instead of simulating for the entire physical operation time which is prohibitively large, we chose to simulate for one minute of operation and project the results (segregation index with time) in time–space using a time-series forecasting method, ARIMA, to overcome the prohibitively large simulation time of a high-fidelity simulation technique like DEM.

ARIMA[26] is one of the most widely used approaches for time-series forecasting in finance[14] and econometrics as it aims to describe the autocorrelation in the data for forecasting. ARIMA models can handle both seasonal

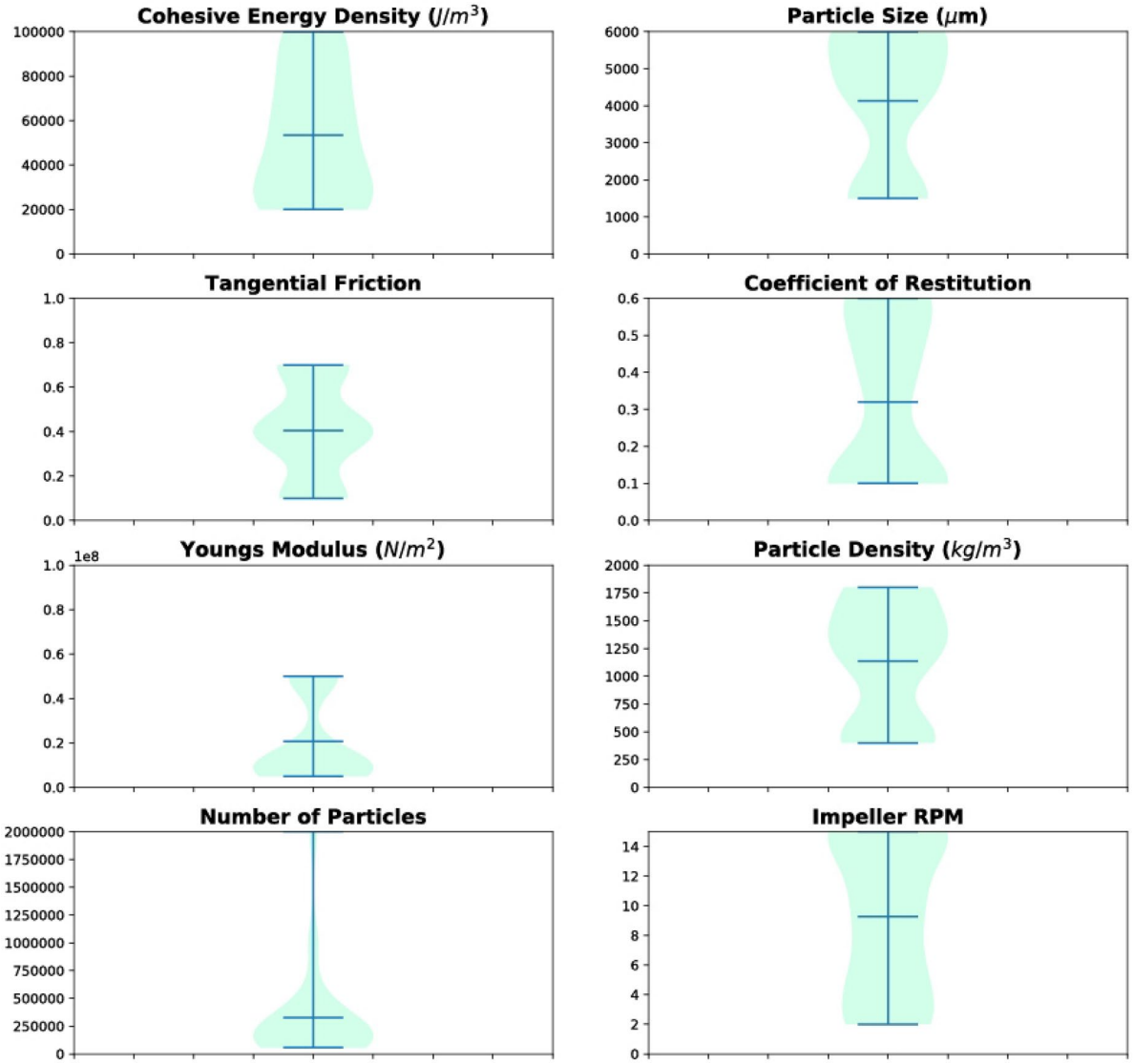

Figure 3. Violin plot of the variation in material properties and process parameters assessed, collectively known as predictor variables. The machine learning model was trained using 8 predictor variables, namely cohesive energy density, particle size, particle size, tangential friction, coefficient of restitution, Young's modulus, particle density, number of particles, and impeller RPM. In each plot, the second horizontal line (out of the three lines) shows the mean value of the individual material property, and the thickness shows the frequency of that particular value across all the simulations. It can be seen that there is good variability in the values of the properties except for number of particles, which can be attributed to the computational challenges of simulating a larger number of particles.

and non-seasonal data and offer advantage over classical exponential smoothing methods. Spline-fitting was also implemented, but it did not perform well due to the noisy nature of the data in certain cases. Time-series data can sometimes be extremely noisy making it difficult to untangle the mean 'stationary' behavior from the noise. ARIMA can transform time-series data into 'stationary' post-differencing, or in other words, a combination of a signal and noise. The elements constituting ARIMA are the number of autoregressive terms required for good forecasting (p), the number of differencing operations to achieve stationarity (d), and the number of lagged forecast errors (q). ARIMA formulation is explained in detail in supplementary section S.2. Differencing and regression using the 'relevant' previous time points, unlike other methods, helped ARIMA to capture the non-seasonal and non-stationary behavior of the segregation index at higher RPMs.

We chose to do time-forecasting of Segregation index, ψ, which is an indicator of the extent of particulate mixing. ARIMA predictions were verified on all DEM generated data by training on $\psi_{t=0}$ to $\psi_{t=T/2}$, where T is the total time step of the DEM simulation and predicting on the latter half ($t = T/2 + 1$ to $t = T$), as can be seen in Fig. 5a. The ARIMA model was able to capture the temporal evolution of the segregation index with an error margin of less than 2.5% from the prediction of DEM simulations. ARIMA validation is summarized in supplementary section S.4.

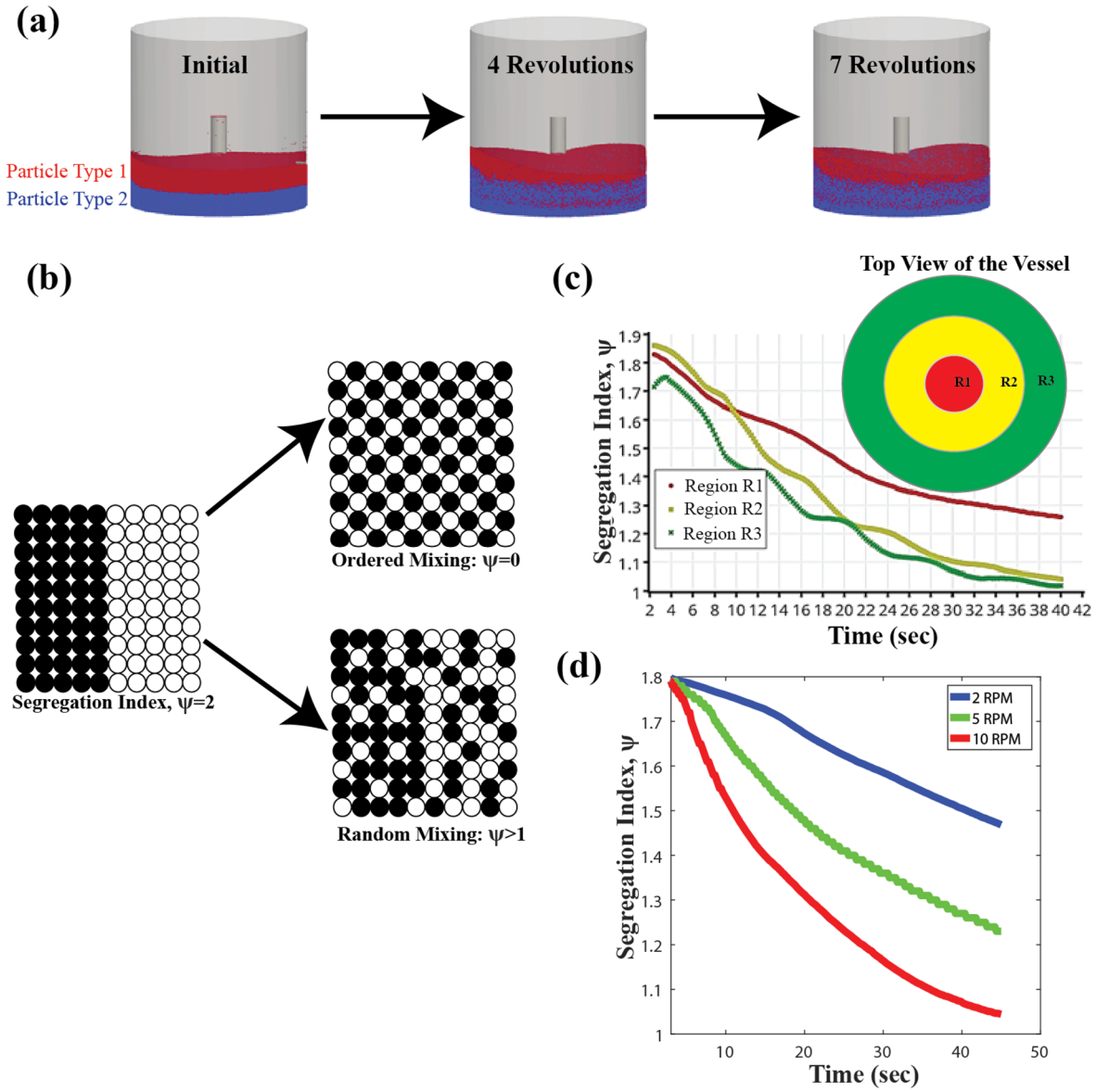

Figure 4. The evolution of segregation index as a function of time and distance from the impeller's axis of rotation. Segregation index was calculated from the spatial positions of the particles at different time steps in the DEM simulations. (**a**) Extent of particle mixing with number of impeller rotations. R = 3 mm, RPM = 15, $E = 5 \times 107$ N/m^2, $\gamma_{cohesion} = 1 \times 105$ J/m^3,$\mu_f = 0.1$, $\rho = 1{,}100$ kg/m^3, e = 0.6, h = 20.33 cm. Particles are labeled by two types to examine their mixing behavior, even though their properties are the same, (**b**) different particle arrangements and the corresponding segregation index[24] (**c**) particle mixing is faster in regions farther from the center of the impeller. Region R1, R2 and R3 span the radial direction of the bed with R1 being the closest to the center of the impeller and R3 being closest to the dryer wall, and (**d**) particle mixing is a function of the number of impeller revolutions. Longer simulations are required for slower RPM.

Post-verification, the ARIMA model was used to forecast the trend of ψ and the time required to reach the desired state of uniform random mixing, i.e. ψ ~ 1 as shown in Fig. 5b. In this work, a cut-off of ψ = 1.1 was chosen to determine uniform mixing as the asymptotically slow approach of ψ towards 1 would result in erroneously large predicted mixing time. It should be noted that the ARIMA model took computational time of *O*(minutes) while DEM simulation would have typically taken another one and half months running on the same computational resources for the results shown in Fig. 5.

ARIMA, though applied here on ψ, could have been applied on another time-varying physical quantity of interest such as torque, stress, and kinetic energy depending on the needs of the study. ARIMA is a powerful tool to reduce the computational cost and time by several orders of magnitude, in terms of core-hours, as indicated in Table 1. End-point estimation using the combination of DEM and ARIMA frees up computational resources that can now be utilized for a parameter sweep of the entire relevant range of material properties and process parameters to build a robust machine learning model.

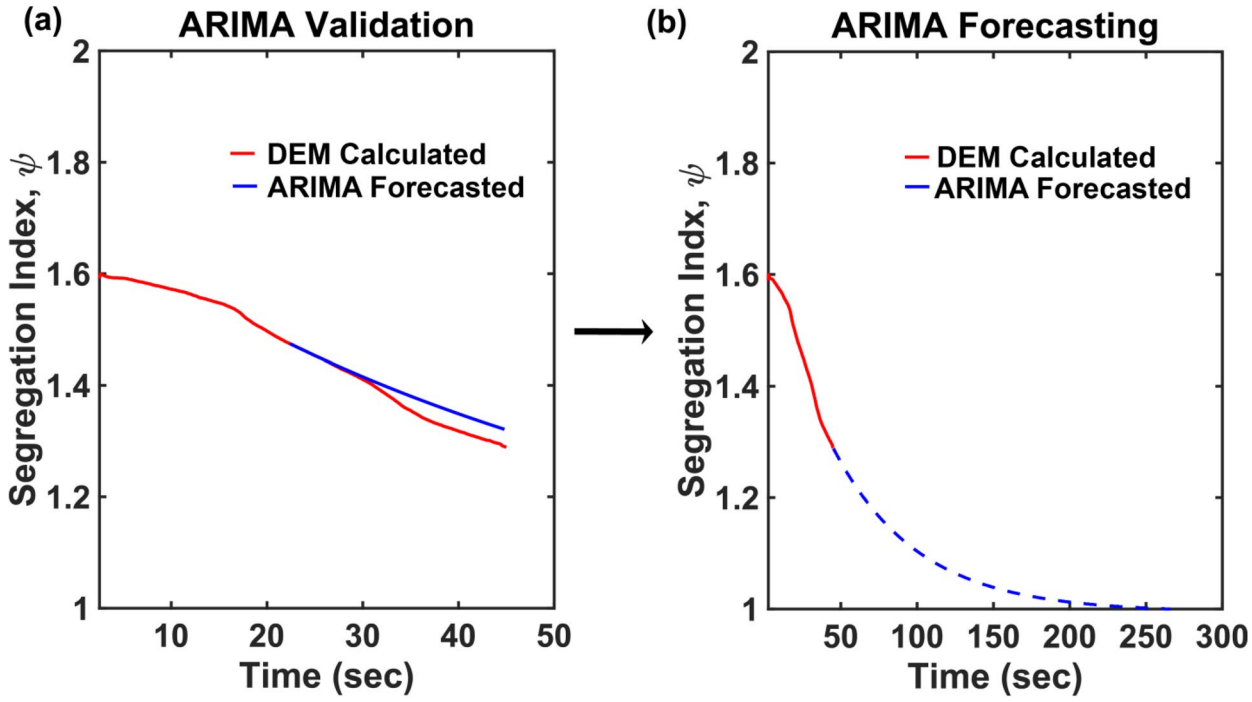

Figure 5. Validation of ARIMA time-series forecasting. (**a**) ARIMA was verified against the DEM simulations for impeller speed of 2 RPM, and (**b**) ARIMA was used to forecast ψ until the bed was uniformly mixed.

Particle size, R (μm)	Number of particles, Np	Fill level, h (cm)	Young's modulus (N/m^2)	DEM simulation time (CPU hours)
1,500	250,000	0.69	1e+7	1584
1,500	1,000,000	2.79	2.5e+7	4,248
4,500	135,000	10.17	5e+7	1,685
4,500	270,000	20.33	5e+6	2,160

Table 1. Computational time of DEM and ARIMA simulations. Computational time for DEM significantly increases with the number of particles, whereas the computational time for ARIMA is only affected by the number of previous time steps to analyze and the number of future time steps to forecast. Hence, the computational time to run ARIMA for each of the cases was on *O*(minutes).

Machine learning predictive model. The desire to develop a ML model stemmed from our vision to utilize data from high-fidelity simulations for process optimization and online control. We envision a manufacturing platform where online advanced process analytical tools (PAT) are feeding process data to a controller which utilizes high-fidelity simulations guided ML models for making process decisions. Once connected with PAT devices, these ML models can improve their prediction over time as more process data becomes available. ML enables learning from a large number of process descriptors along with advanced feature engineering, which leads to robust predictions of systems with complex phenomena, and also has been shown to be more accurate than linear regression techniques[6,27,28]. Another advantage of a ML model is that it eliminates the need for running costly high-fidelity simulations in the future and provides deeper insights and patterns which were otherwise not easy to decipher. Although, machine learning methods are great at predicting interpolated results, they may not perform well when the values for the descriptors are far from the training set. To overcome this limitation, we created a diverse descriptor design space.

In this work, ARIMA forecasted uniform mixing time was taken as the response variable to be predicted as a function of a set of input parameters such as material properties and process parameters. Implementing sophisticated machine learning methods, such as neural networks, was tempting but not practical because of the dimensions of the dataset making it vulnerable to over-fitting. Random forest outperformed ($R^2 = 0.79$) the other methods because of averaging the results from multiple trees generated from a randomly selected subset of the data. Partial least squares regression (PLSR), support vector regression, and regularized linear regression technique Elastic Net were inferior in performance as compared to random forest with an R^2 of 0.70, 0.72 and 0.71 respectively, also can be seen in Fig. 6. However, all these methods performed better than the conventional linear regression because of the non-linear interactions arising from the complex interplay of the underlying multi-physics phenomena. Leave-one-out cross-validation was performed on all the above investigated ML

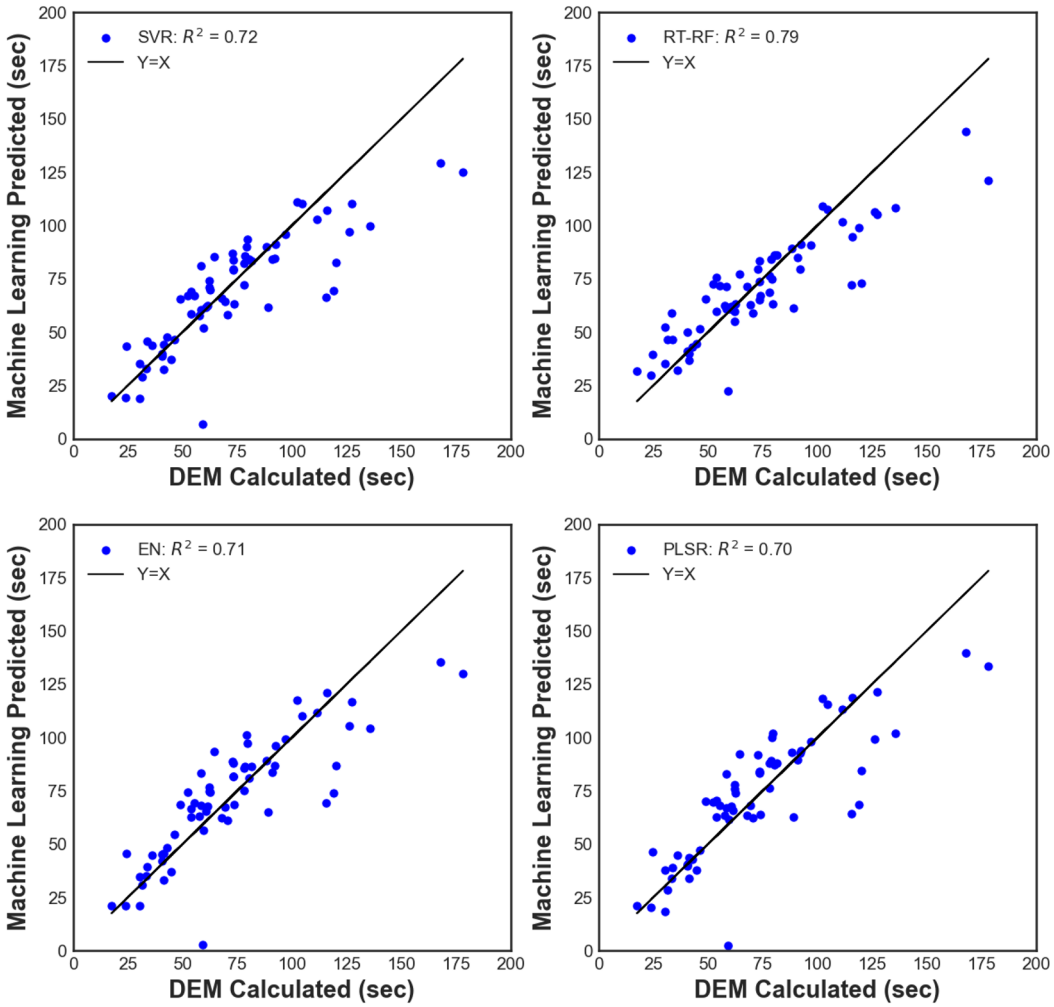

Figure 6. ML prediction of agitation time compared to DEM-ARIMA simulations. Leave-one-out cross-validation was performed to evaluate the methods. RT-RF performed the best amongst all the methods with an R^2 of 0.79.

methods to test their prediction and also vulnerability to over-fitting. Further, robustness in performance can be ensured as more and more process data becomes available for integration into the existing ML models.

Having obtained the predictive ML model, we sought to gain mechanistic insight in the system (an agitated filter dryer) by probing which descriptors impact the response variable the most by a process known as feature selection. RT-RF, our best ML model, identifies the importance of the descriptors rather than weights attached to the descriptors like in a linear regression model. Importance of a feature is quantified by calculating the percentage change in mean-squared error by changing the value of the descriptor. According to RT-RF, fill level, impeller rotation, and particle radius, in decreasing order of significance, are the most informative descriptors to impact uniform mixing time which is in good agreement with some recent works[29,30].

At larger fill levels, particles need to be displaced to a greater extent to achieve uniform mixing leading to an increase in mixing time[29]. In a similar manner, at the same fill level, increasing impeller speed creates larger convective diffusion and thus reduces mixing time[29]. Local shear diffusion rate scales as $\sim \dot{\gamma} a^2$, where $\dot{\gamma}$ is the local shear rate and a is the particle radius, which was also identified as an important parameter by random forest. We hypothesize that, in our system, convective and shear diffusion play an important role in mixing based on these results. Though similar conclusions could have been arrived by other means, ML allows us to provide relative weight to each descriptor of the system and thus provides a framework for mechanistic exploration. In a convoluted system, like the one studied here, where there are multiple descriptors and fundamental understanding is missing, ML can be a powerful tool to point theorists in the right direction.

Discussion

A large amount of resources and time are spent in a variety of industries dealing with solid handling in developing a robust, scalable, and reproducible process. Fundamental scientific tools, though accurate, have prohibitively large computational cost, particularly for transient cases, while most industrial processes, either batch or

continuous, have a transient component in their operation. The study presented here shows that for a complex and relevant case of cohesive powder mixing, a novel approach based on time-series forecasting using ARIMA and ML can provide tremendous insights and guide development of a mechanistic framework that identifies key descriptors that most significantly impact the process. The overall framework presented here is quite simple and powerful and can be adopted in a variety of engineering and scientific problems that are transient in nature. A simpler extension of the framework can be done in the field of computational fluid dynamics (CFD) to probe various heat and mass transfer limited and phase transition systems. Similarly, it can be used in the field of molecular dynamics (MD) to predict the molecular structures of materials, biological entities like DNA or proteins. Coupling of fundamental tools with ARIMA and ML reduce the computational time to probe a large descriptor set and provide predictions on the behavior of the system under new conditions and/or, additionally, the optimal way of operating the system. From an industrial perspective, ML models can become part of model-predictive control and, coupled with PAT and automation, they can provide endless opportunities.

Method description

DEM simulation. SmartDEM (Tridiagonal Solutions, San Antonio, TX) software was employed to perform all DEM simulations. SmartDEM is a GUI implementation of the open-source DEM code LIGGGHTS (LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) Improved for General Granular and Granular Heat Transfer Simulations; CFDEM Project) and allows for ease of simulation setup and result interpretation. Multiple automated scripts were written to create different simulation setup, submit jobs to a supercomputing cluster, and post-process the gigabytes of simulated data. The details of the DEM formulation are in supplementary section S.1.

ARIMA forecasting. A python code[31] was customized to forecast the segregation index. As the mixing time is a complex function of the descriptors, one ARIMA model would not work best for all the data. ARIMA hyperparameters (p,d,q) were therefore sampled between 0 and 100, 0 and 2, 0 and 2 respectively for all the simulations. Given that errors at previous time steps are unobserved variables, maximum likelihood estimation (MLE) was performed in order to find the best model. Akaike information criterion[32] (AIC) score was used to select the best ARIMA model after comparing each model against other models. ARIMA python codes were run on the Anaconda[33] platform using jupyter notebooks. All the simulations took O(minutes) for completion, which reflects on the power and scalability of the method. The time complexity of ARIMA is a function of the number of values of hyperparameters to sample rather than the number of particles or the values of the other descriptors, as compared to the DEM simulations where computational time significantly depends on the fill level and the number of particles.

Machine learning methods. Machine learning methods such as elastic net regression (EN)[34], support vector regression (SVR)[35], partial least squares regression (PLSR)[36], and regression tree random forest (RT-RF)[37] were used to build the predictive model (refer to Fig. 2). Due to the limited number of datasets available, artificial neural networks was not implemented because of the concerns of over-fitting. A variety of linear, regularized linear, and non-linear methods were evaluated, of which random forest performed the best. Leave-one-out cross-validation was performed to evaluate the machine learning methods and test the vulnerability of the methods towards over-fitting. Hyperparameter tuning for all the machine learning methods was done using the GridSearchCV option in scikit-learn[38]. Hyperparameters for random forest such as number of descriptors and maximum depth of each tree were sampled and bootstrapping was permitted. The computational time for running the machine learning methods were on O(mins), which is astronomically lower than the alternative option of DEM simulations, which would have taken months of computational time.

Received: 23 April 2019; Accepted: 5 June 2020
Published online: 13 July 2020

References

1. Vafaee, F. *et al.* A data-driven, knowledge-based approach to biomarker discovery: application to circulating microRNA markers of colorectal cancer prognosis. *NPJ Syst. Biol. Appl.* **4**(1), 1–12 (2018).
2. Oliver, N. M., Rosario, B. & Pentland, A. P. A Bayesian computer vision system for modeling human interactions. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 831–843 (2000).
3. Collobert, R. & Weston, J. A unified architecture for natural language processing. In *Proceedings of the 25th International Conference on Machine Learning—ICML '08* 160–167 (ACM Press, 2008).
4. Bojarski, M. *et al.* End to end learning for self-driving cars. Preprint at https://arxiv.org/abs/1604.07316 (2016).
5. Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications. *Drug Discov. Today* **20**, 318–331 (2015).
6. Brockherde, F. *et al.* Bypassing the Kohn-Sham equations with machine learning. *Nat. Commun.* **8**, 872 (2017).
7. Kresse, G. & Hafner, J. *Ab initio* molecular dynamics for liquid metals. *Phys. Rev. B* **47**, 558–561 (1993).
8. Wendt, J. F. (ed.) *Computational fluid dynamics: an introduction* (Springer, Berlin, 2008).
9. Ayers, P. W. & Yang, W. Density functional theory in Computational medicinal chemistry for drug discovery (eds Bultinck, P., de Winter, H., Langenaeker, W. & Tollenaere, J. P.) 571–616 (CRC Press, 2003).
10. Munjiza, A. *The combined finite-discrete element method* (Wiley, Hoboken, 2004).
11. Hughes, T. J. R. *The finite element method: linear static and dynamic finite element analysis* (Dover Publications Inc., Mineola, 2000).
12. Bezzo, F., Macchietto, S. & Pantelides, C. C. General hybrid multizonal/CFD approach for bioreactor modeling. *AIChE J.* **49**, 2133–2148 (2003).
13. Vrábel, P. *et al.* CMA: integration of fluid dynamics and microbial kinetics in modelling of large-scale fermentations. *Chem. Eng. J.* **84**, 463–474 (2001).
14. Pai, P.-F. & Lin, C.-S. A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega* **33**, 497–505 (2005).
15. Cooke, M. H., Stephens, D. J. & Bridgwater, J. Powder mixing—a literature survey. *Powder Technol.* **15**, 1–20 (1976).

16. Poux, M., Fayolle, P., Bertrand, J., Bridoux, D. & Bousquet, J. Powder mixing: some practical rules applied to agitated systems. *Powder Technol.* **68**(3), 213–234 (1991).
17. Sen, M. & Ramachandran, R. A multi-dimensional population balance model approach to continuous powder mixing processes. *Adv. Powder Technol.* **24**, 51–59 (2013).
18. Sen, M., Dubey, A., Singh, R. & Ramachandran, R. Mathematical development and comparison of a hybrid PBM-DEM description of a continuous powder mixing process. *J. Powder Technol.* **2013**, 1–11 (2013).
19. Chaudhuri, B., Mehrotra, A., Muzzio, F. J. & Tomassone, M. S. Cohesive effects in powder mixing in a tumbling blender. *Powder Technol.* **165**, 105–114 (2006).
20. Conder, E. W. *et al.* The pharmaceutical drying unit operation: an industry perspective on advancing the science and development approach for scale-up and technology transfer. *Org. Process Res. Dev.* **21**, 420–429 (2017).
21. Bridgwater, J. Fundamental powder mixing mechanisms. *Powder Technol.* **15**, 215–236 (1976).
22. Birch, M. & Marziano, I. Understanding and avoidance of agglomeration during drying processes: a case study. *Org. Process Res. Dev.* **17**, 1359–1366 (2013).
23. Hoffmann, H. Simple violin plot using matlab default kernel density estimation. https://www.mathworks.com/matlabcentral/fileexchange/45134-violinplot (2015).
24. Marigo, M., Cairns, D. L., Davies, M., Ingram, A. & Stitt, E. H. A numerical comparison of mixing efficiencies of solids in a cylindrical vessel subject to a range of motions. *Powder Technol.* **217**, 540–547 (2012).
25. Hare, C., Ghadiri, M. & Dennehy, R. Prediction of attrition in agitated particle beds. *Chem. Eng. Sci.* **66**, 4757–4770 (2011).
26. Box, G. E. P., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. *Time series analysis: forecasting and control* (Wiley, Hoboken, 2016).
27. Huynh-Thu, V. A., Irrthum, A., Wehenkel, L. & Geurts, P. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE* **5**, e12776 (2010).
28. Ahneman, D. T., Estrada, J. G., Lin, S., Dreher, S. D. & Doyle, A. G. Predicting reaction performance in C-N cross-coupling using machine learning. *Science* **360**, 186–190 (2018).
29. Dubey, A., Sarkar, A., Ierapetritou, M., Wassgren, C. R. & Muzzio, F. J. Computational approaches for studying the granular dynamics of continuous blending processes, 1-DEM based methods. *Macromol. Mater. Eng.* **296**, 290–307 (2011).
30. Liu, Y., Gonzalez, M., Wassgren, C. & Gonzalez, M. Modeling granular material blending in a rotating drum using a finite element method and advection-diffusion equation multi-scale model. *AIChE J.* **64**, 3277–3292 (2018).
31. Vincent, T. ARIMA time series data forecasting and visualization in python. DigitalOcean. https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting-with-arima-in-python-3 (2017).
32. Bozdogan, H. Model selection and akaike's information criterion (AIC): the general theory and its analytical extensions. *Psychometrika* **52**, 345–370 (1987).
33. Continuum Analytics. Anaconda software distribution. Computer software Vers. 2-2.4.0 (2016).
34. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B. (Stat. Methodol.)* **67**, 301–320 (2005).
35. Basak, D., Pal, S. & Patranabis, D. C. Support vector regression. *Neural Inf. Process. Lett. Rev.* **11**(10), 203–224 (2007).
36. Geladi, P. & Kowalski, B. R. Partial least-squares regression: a tutorial. *Anal. Chim. Acta* **185**, 1–17 (1986).
37. Svetnik, V. *et al.* Random forest: a classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.* **43**(6), 1947–1958 (2003).
38. Pedregosa, F. *et al.* Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

Author contributions

P.K. (lead author) worked on the machine learning and time-series analysis of the study. He also ran and analyzed the DEM simulations. Prashant worked on drafting the manuscript and generated all the figures. K.S. (corresponding author) conceived, designed and oversaw the entire work, ran DEM simulations, and worked on drafting the manuscript. N.N. and L.M. helped in designing and guiding the work by providing practical insights from their operational experiences which guided the DEM simulation work. They also reviewed the manuscript, and substantially revised it. Y.S. and R.H. helped in designing and guiding the work by providing insight in range of material properties and their behavior in pharmaceutical operations which guided DEM simulations and subsequent ML work. They also reviewed the manuscript and substantially revised it. A.S. reviewed the manuscript and helped in its restructuring and revisions.

Competing interests

All authors except Prashant Kumar and Yujin Shin (prior AbbVie employees) are AbbVie employees and may own AbbVie stock. Prashant Kumar is currently an employee of Analysis Group and Yujin Shin is currently an employee of Abbott Laboratories. Prashant Kumar and Yujin Shin were both AbbVie employees when the work was performed and have no additional conflict of interest to disclose. AbbVie sponsored and funded the study; contributed to the design; participated in the collection, analysis, and interpretation of data, and in writing, reviewing, and approval of the final manuscript. The author(s) declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41598-020-67546-w.

Correspondence and requests for materials should be addressed to K.S.

Reprints and permissions information is available at www.nature.com/reprints.

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