Physics-informed Machine Learning for Accelerated Testing of Roll-to-roll Printed Sensors

S Chandra Mouli
Department of Computer Science
Purdue University

Sotoudeh Sedaghat
School of Materials Engineering
Purdue University

Muhammed Ramazan Oduncu
School of Materials Engineering
Purdue University

Ajanta Saha
School of Electrical and Computer Engineering
Purdue University

Rahim Rahimi
School of Materials Engineering
Purdue University

Muhammad A. Alam
School of Electrical and Computer Engineering
Purdue University

Alexander Wei
School of Materials Engineering
Purdue University

Ali Shakouri
School of Electrical and Computer Engineering
Purdue University

Bruno Ribeiro
Department of Computer Science
Purdue University

Roll-to-roll printing has significantly shortened the time from design to production of sensors and IoT devices, while being cost-effective for mass production. But due to less manufacturing tolerance controls available, properties such as sensor thickness, composition, roughness, etc., cannot be precisely controlled. Since these properties likely affect the sensor behavior, roll-to-roll printed sensors require validation testing before they can be deployed in the field. In this work, we improve the testing of Nitrate sensors that need to be calibrated in a solution of known Nitrate concentration for around 1-2 days. To accelerate this process, we observe the initial behavior of the sensors for a few hours, and use a physics-informed machine learning method to predict their measurements 24 hours in the future, thus saving valuable time and testing resources. Due to the variability in roll-to-roll printing, this prediction task requires models that are robust to changes in properties of the new test sensors. We show that existing methods fail at this task and describe a physics-informed machine learning method that improves the prediction robustness to different testing conditions ($\approx 1.7\times$ lower in real-world data and $\approx 5\times$ lower in synthetic data when compared with the current state-of-the-art physics-informed machine learning method).

Keywords: *physics-informed machine learning, roll-to-roll printing, out-of-distribution*

1 INTRODUCTION

Roll-to-roll printing is being widely used for manufacturing a variety of flexible electronics including sensors, wearable implants, capacitors etc., especially because of the reduced large-scale manufacturing cost. However, unlike traditional high-cost manufacturing processes, roll-to-roll printing does not allow precise control over parameters such as thickness, chemical composition of sensor membrane, and roughness/resistance of electrodes. These properties can impact the sensors' response, resulting in a significant sensor-to-sensor variability making precise measurements much harder. Thus, roll-to-roll printed sensors typically require validation testing before they can be deployed in the field. We use machine learning approaches to predict the behavior of roll-to-roll printed sensors at the end of validation testing after observing only their initial measurements for a short period of time. This way, we aim to *accelerate* the pre-deployment validation of these printed sensors.

In this work, we consider potentiometric ion-selective electrodes [10] that can measure the nitrate concentration in water and soil. These nitrate sensors are integral for real-time soil health monitoring for precision agriculture. However, before deployment, these sensors require a conditioning phase when they are kept in a solution of known nitrate concentration until saturation (can take around 1-2 days). To accelerate this process, we wish to observe the initial behavior of the sensors for a few hours, and predict the future measurements at the end of conditioning.

Standard data-driven machine learning (ML) methods have shown a remarkable ability to fit the data and could be used for this task. However, this ability can come at the expense of lack of robustness to changes to the input data, for example, when the test conditions are different from those in training. This is known as an *out-of-distribution* task: when the test data and the training data are not from the same distribution. In our application, an out-of-distribution scenario occurs when the test sensors have very different membrane thickness when compared to the sensors seen during training. ML methods tend not to perform well under such out-of-distribution scenarios as they may learn shortcuts [6], simple but highly predictive correlations between the known inputs and the desirable outputs as seen in the training data. This ability is orthogonal to the concept of model overfitting: An ML model that does not overfit can still rely on spurious correlations for its predictions. Physical models on the other hand can calibrate their parameters to an out-of-distribution test condition and are able to extrapolate; however, they do not model all the complex real-world processes and hence, do not fit the data well enough to forecast accurately.

Physics-informed machine learning (PIML) has emerged as a hybrid solution and positively impacted a diverse set of fields over the past few years including biological sciences [22], climate science [4], turbulence modeling [12, 20], etc. The works in this field incorporate domain knowledge in the form of physics modeling into standard machine learning models. The main goal of PIML works is to improve standard machine learning models so that they **(a)** produce physically-consistent results [11], **(b)** learn from less data, and **(c)** can make better predictions in *out-of-distribution tasks*, i.e., predict in new conditions not seen in the training data.

We show that PIML methods have inherited the out-of-distribution weakness from standard machine learning models and cannot extrapolate in our application. Then, we describe a physics-informed machine learning model that is less sensitive to out-of-distribution tasks.

2 RELATED WORK

There are three standard ways of incorporating physics-based constraints in machine learning models:

Learning physics constraints from data. Schmidt and Lipson [19] propose a genetic algorithm to learn natural laws such as Hamiltonian of a system, equations of motion, etc., in the form of invariances solely from experimental data. More recent works [1, 13, 17] learn equations (explicit solution or partial differential equations) by regressing over a dictionary of basis functions (e.g., sin, cos, $\frac{d}{dt}$, etc.) and use sparsity constraints to ensure learning of simpler laws.

Hard constraints. Many physics-informed ML methods enforce the constraints strictly in the architecture. [18, 9] incorporate a known PDE in neural networks as a hard constraint and learn the unknown parameters via supervised learning. Works have strictly incorporated energy conservation by learning the Hamiltonian/Lagrangian [7, 3] of the given systems. Other works have incorporated group symmetries such as translation and rotation [21, 5]. The advantage of embedding hard constraints is that the constraints continue to be satisfied even outside the training domain. However, this requires one to know all the constraints of the system precisely; if the constraints do not hold or are misspecified, then we cannot recover from an incorrect choice.

Soft constraints. Rather than enforcing physics-informed constraints strictly in the architecture, many works prefer to use a "soft" regularization of these constraints [11, 16, 8]. This typically allows the neural networks to violate the constraints if absolutely required in training, thus allowing for noisy data. However, "soft" constraints also have the drawback that they are typically not satisfied outside the training domain.

3 PROBLEM STATEMENT

In this section we formally describe the out-of-distribution task. Many dynamical systems can be written as an ordinary differential equation (ODE) with $\mathbf{X}(t)$ describing the state of the system at time t:

$$\frac{d\mathbf{X}(t)}{dt} = \psi(t, \mathbf{X}(t)) \,, \tag{1}$$

where $\mathbf{X}(t) \in \mathbb{R}^d$, and ψ is a deterministic function. We will restrict our attention to discrete time steps $\{t_0, \ldots, t_T\}$. Given an initial value of the system $\mathbf{X}(t_0)$ at time $t = t_0$, one can solve the ODE in Equation (1) for the future states of the system for $t > t_0$. However, since ψ is unknown to us, we wish to learn a machine learning model that forecasts the future values of the dynamical system for $t > t_r$ when given initial observations for $t \le t_r$ as input. For ease of notation, we will denote the past observations by $\mathbf{X}_{\le r} \equiv \mathbf{X}(t_0), \ldots, \mathbf{X}(t_r)$ and the future observations by $\mathbf{X}_{>r} \equiv \mathbf{X}(t_{r+1}), \ldots, \mathbf{X}(t_T)$.

Physics model. We assume that we are given a physics model $d\mathbf{X}(t)/dt = \phi(t, \mathbf{X}(t); \theta_{\text{phy}})$ that approximately describes the dynamical process ψ, where θ_{phy} are parameters that can be calibrated. Typically, ϕ is much simpler than the real dynamical process ψ, and data-driven approaches are needed to complement it for better predictions.

Data-driven approach. The training data for the data-driven approaches consists of multiple simulations of the dynamical system in Equation (1) with different initial conditions $\mathbf{X}(t_0)$. We denote our training data $\mathcal{D}^{(\text{tr})} = \{(\boldsymbol{x}^{(i)}_{\le r}, \boldsymbol{x}^{(i)}_{>r})\}_{i=1}^{N^{(\text{tr})}}$. After an ML model is trained on the training data, at test time, we ask the question: what if the initial conditions were different from those in training? We call this the *out-of-distribution* test data. In our application, one training sample consists of voltage measurements of a nitrate sensor during its conditioning phase. Out-of-distribution test data consists of voltage measurements from sensors that were manufactured with different control parameters.

While standard data-driven approaches are adept at making accurate predictions when test is similar to the training distribution, they fail to extrapolate to the out-of-distribution datasets. Surprisingly, the physics-informed machine learning models have inherited this drawback from the data-driven approaches. The goal of the current work is to train a physics-informed machine learning model on the training data in such a way that it can extrapolate to out-of-distribution test datasets.

In the next section, we describe our application in detail.

4 APPLICATION: ISE NITRATE SENSOR CONDITIONING

Potentiometric ion-selective electrodes (ISEs) are being adopted as implantable sensors for precision agriculture, for instance to measure nitrate concentrations in the soil. This adoption has predominantly been due to their easy and low-cost fabrication via roll-to-roll printing. Potentiometric sensors are 2-electrode systems that are comprised of working and reference electrodes. The working electrode (ISE) is constructed with a selective plasticized membrane deposited onto a conductive solid contact. Activity of the ions in the solution results in a potential change on the ISE that is measured with respect to the reference electrode. Thus, target ion activities are translated into potential readings providing an estimate of the ion concentration in the solution. Nernst equation can be used to obtain the ion concentration from the measured potential difference. In our application, we measure the nitrate concentration by using a nitrate ion-selective membrane coated on a printed silver working electrode.

The current usage of ISEs is not without its challenges. These sensors require a period of preconditioning where the ISEs are activated by hydrating the ion selective membrane in a standard solution of known analyte concentration before deployment in the field. They could take around 1-2 days to reach a constant saturation voltage (thermodynamic equilibrium). Once the sensors are preconditioned, any change in the nitrate concentration is detected in a matter of seconds. Calibration of hundreds of sensors is time-consuming and resource-intensive. Our task then is to predict the saturation voltage at the end of the conditioning period given a small initial period of transient-state voltage measurements. However, the saturation voltage depends upon the manufacturing conditions that can vary for different sensors. As we show later, thickness of a sensor's ion-selective membrane has a significant impact on its saturation voltage. We need to predict the saturation voltage of a test sensor under this out-of-distribution scenario.

4.1 Real-world data

We evaluated the electrochemical performance of the ISEs manufactured in different coating runs using Multi-

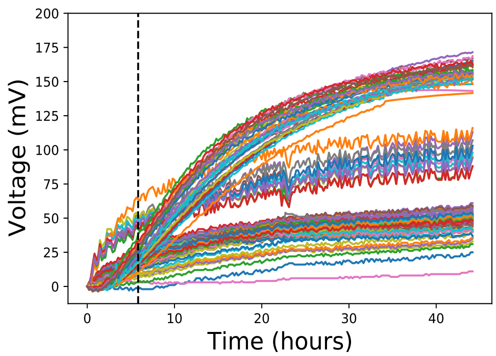

Fig. 1: Potentiometric Nitrate sensors conditioned in 1 mM Nitrate solution until saturation. We wish to predict the saturation behavior given first 6 hours of voltage readings.

function DAQ system (National Instruments PXI-6225) equipped with a LabView system engineering software. We collected the potential outputs of around 300 electrodes against a commercial Ag/AgCl reference electrode (Orion 900200 Sure-Flow) with the ISEs placed in 1L of 1mM potassium nitrate solution until equilibrium. The resultant voltage readings for a sample of the sensors are shown in Figure 1. We can see that the sensors take roughly 40 hours to reach a constant saturation voltage. Our task then is to predict the saturation voltage given a small initial period of voltage measurements (left of dashed vertical line in Figure 1). Further, since the sensors were manufactured with different control parameters, they have very different saturation voltages even when placed in a solution of same nitrate concentration. Thus, our goal is to design methods that can extrapolate and forecast robustly under changes in manufacturing parameters.

4.2 Synthetic data

In the following, we perform controlled out-of-distribution experiments to evaluate the different forecasting approaches. This way, we are able to ensure that only certain manufacturing conditions change between the training and the test data. We generate synthetic voltage curves from the following model proposed by Jin et al. [10] that describes the conditioning behavior of the potentiometric nitrate sensors:

$$V_{\text{trans}}(t) = \frac{k_B T}{q} \ln(t) + C_1\sqrt{Dt} + C_2 \,, \tag{2}$$

$$V_{\text{sat}} = \frac{k_B T}{q} \ln\left(\frac{n_0 h^2}{\kappa}\right) + C_3 \,, \tag{3}$$

where $V_{\text{trans}}(t)$ denotes the transient voltage readings and V_{sat} is the constant saturation voltage (independent of time). In Equations (2) and (3), k_B is the Boltzmann's constant, q is the elementary charge, T is the temperature of the nitrate solution, D is the diffusivity of the nitrate ion, and κ is the dielectric constant of the ion-selective membrane. In Equation (3), n_0 denotes the nitrate concentration of the solution, and h denotes the thickness of the sensor membrane. The saturation voltage is proportional to the logarithm of h^2, which is unknown to the prediction methods.

The final voltage reading is an interpolation of the transient V_{trans} and saturation voltage V_{sat}, given by

$$V(t) = \frac{V_{\text{trans}}(t)}{\left(1 + \left(\frac{V_{\text{trans}}(t)}{V_{\text{sat}}}\right)^{\beta}\right)^{\frac{1}{\beta}}} \,, \tag{4}$$

where we choose $\beta = 1.5$ in our experiments.

Training data and out-of-distribution test data. We obtain voltage curves from Equation (4) with additive Gaussian noise and fixed values for the Nitrate concentration $n_0 = 10^{-3}$M and temperature $T = 298K$. We keep these quantities fixed because these can typically be controlled during the conditioning phase. However, since there is little control over the membrane thickness h in roll-to-roll printing, we sample a range of different values for sensor thickness h. For training data, we obtain voltage curves with sensor thickness between 50 to 60 microns. Then, we simulate the case when the manufacturing control parameters change for the out-of-distribution test data by sampling much higher sensor membrane thickness between 150 to 160 micros. The sensors in out-of-distribution test data have higher saturation voltage than the ones in the training data.

4.3 Physics model for forecasting

In the next section, we describe a physics-informed machine learning approach that improves the out-of-distribution robustness of existing methods. We use the following *simpler* physics model in our application [10]:

$$\begin{aligned}\varphi_{\text{trans}}(t) &= \frac{k_B T}{q}\ln(t) + C_0 \,, \\ \varphi_{\text{sat}} &= \frac{k_B T}{q}\ln\left(\frac{n_0\theta_h^2}{\kappa}\right) + \theta_C \,, \\ \varphi(t) &= \frac{\varphi_{\text{trans}}(t)}{\left(1+\left(\frac{\varphi_{\text{trans}}(t)}{\varphi_{\text{sat}}}\right)^{\theta_\beta}\right)^{\frac{1}{\theta_\beta}}} \,, \end{aligned} \tag{5}$$

where $\theta_{\text{phy}} = (\theta_h, \theta_C, \theta_\beta)$ are the parameters of the physics model and θ_h directly corresponds to the sensor thickness h in Equation (3). The model does not account for effects of water in the membrane and thus, ϕ_{trans} does not involve the $\sqrt{Dt}$ term from Equation (2). While the physics model in Equation (5) is given in closed form, we can easily rewrite it as an ordinary differential equation $\frac{d\varphi}{dt} = \phi(t; \theta_{\text{phy}})$ (as required by the physics-informed machine learning approaches we consider).

5 PROPOSED APPROACH

We begin with a description of a state-of-the-art physics-informed machine learning framework, APHYNITY [8].

5.1 APHYNITY

APHYNITY describes the prediction dynamics using an ordinary differential equation (ODE),

$$\begin{aligned}\frac{d\hat{\mathbf{X}}(t)}{dt} &= \phi(t, \hat{\mathbf{X}}(t); \hat{\theta}_{\text{phy}}) + F_{\text{nn}}(\hat{\mathbf{X}}(t); \boldsymbol{W}_F) \,, \\ \hat{\theta}_{\text{phy}} &= R_{\text{nn}}(\mathbf{X}_{\leq r}; \boldsymbol{W}_R) \,, \end{aligned} \tag{6}$$

where $\hat{\mathbf{X}}(t)$ denote the predictions, F_{nn} is a feedforward neural network with learnable weights $\boldsymbol{W}_F$, and ϕ is the known physics model with parameters $\hat{\theta}_{\text{phy}}$ that are predicted via a recurrent neural network R_{nn} with weights $\boldsymbol{W}_R$. Essentially, the neural network F_{nn} acts as a correction term on top of the physics model ϕ. Given the initial prediction at time $t = t_0$ as $\hat{\mathbf{X}}(t_0) = \mathbf{X}(t_0)$, the ODE is solved to obtain the predictions at future times $\hat{\mathbf{X}}(t_1), \ldots, \hat{\mathbf{X}}(t_T)$.

Given the training data $\mathcal{D}^{(\text{tr})}$, the method learns $\boldsymbol{W}_F, \boldsymbol{W}_R$ and θ_{phy} by solving the following optimization problem

$$\begin{aligned}&\underset{\boldsymbol{W}_F, \boldsymbol{W}_R, \theta_{\text{phy}}}{\text{minimize}} \sum_{i=1}^{N^{(\text{tr})}} \sum_{k=0}^{T} ||F_{\text{nn}}(\boldsymbol{x}_k^{(i)}; \boldsymbol{W}_F)||^2 \\ &\text{s.t. } \hat{\boldsymbol{x}}_k^{(i)} = \boldsymbol{x}_k^{(i)}, \forall i = 1, \ldots, N^{(\text{tr})}, \forall k = 0, \ldots, T \,, \end{aligned} \tag{7}$$

where $\hat{\boldsymbol{x}}_k^{(i)}$ for time t_k is obtained by solving the ODE in Equation (6) with the initial value $\boldsymbol{x}_0^{(i)}$ at time $t = t_0$. Note that in Equation (7), the predictions $\hat{x}_k^{(i)}$ depend on the parameters $\boldsymbol{W}_F, \boldsymbol{W}_R, \theta_{\text{phy}}$.

In words, APHYNITY minimizes norm of the outputs from the neural network F_{nn} (the correction terms) with the constraint that the predictions $\hat{\mathbf{X}}_k^{(i)}$ match the corresponding training curve. In this way, the method is able to give higher preference to the physics model and only use the neural network if necessary. For example, if the physics model ϕ is able to perfectly fit the training data, then the neural network outputs are forced to be zero. The optimization in Equation (7) can be solved via gradient descent with a sequence of Lagrangian relaxations to enforce the constraints [8].

5.2 PhysicsDNA

The APHYNITY model as described above works wonderfully in-distribution (i.e., when the test and training data are sampled from the same distribution), but faces several out-of-distribution challenges. We describe these challenges next and show how PhysicsDNA [15] is able to solve them and output more robust predictions.

Physics parameter prediction. In our application, the calibration parameters $\theta_{\text{phy}} = (\theta_h, \theta_\beta, \theta_C)$ of the physics model can be different for each curve because of the variations in the sensor membrane thickness h. As described above, APHYNITY uses a recurrent neural network R_{nn} to predict the physics model parameters and obtain $\hat{\theta}_{\text{phy}}$ (see Equation (6)). In out-of-distribution scenario with very different manufacturing parameters (e.g., in our application, sensor thicknesses in test are much higher compared that in training), the prediction of physics parameters by a neural network is not robust. Further, the physics model may be very sensitive to some parameter; predicting it with a neural network out of distribution could result in very strange final predictions. In some cases, the neural network might even output physically meaningless values for the physics parameters.

Instead, PhysicsDNA calibrates the parameters θ_{phy} of the physics model by fitting each individual voltage curve from $t_0, \ldots, t_r$ using the Levenberg–Marquardt algorithm. This allows us to obtain the best fitted physics parameter even for the out-of-distribution observations. Note however that these fits may not be perfect as they only consider the initial observations till time t_r (since $t > t_r$ is to be forecasted) and the physics model does not capture all the transient behaviour. Thus, we need a neural network to correct the physics model same as in Equation (6).

Historical observations. The feedforward neural network F_{nn} in Equation (6) does not use the history of input observations $\mathbf{X}(t_0), \ldots, \mathbf{X}(t_r)$. Only input to F_{nn} is $\mathbf{X}(t_0)$ as the initial value for the ordinary differential equation. In our application, $\mathbf{X}(t_0)$ denotes the initial voltage measurement which is 0mV in both in-distribution and out-of-distribution data. On the other hand, the rest of voltage measurements $\mathbf{X}(t_1), \ldots, \mathbf{X}(t_r)$ change between train and test. Thus, the feedforward neural network is not able to even detect the change in input distribution.

Instead, PhysicsDNA proposes to use a sequence model, e.g., a recurrent neural network, instead of the feedforward network in Equation (6). This is similar to the method of Mehta et al. [14]; however, their objective is different from Equation (7) and do not minimize the contributions of the neural network.

6 RESULTS

We evaluate four approaches for the out-of-distribution forecasting in real-world and synthetic datasets: **(a)** a standalone physics model (Equation (5)), **(b)** NeuralODE [2], a standalone deep learning method, **(c)** APHYNITY (Equation (6)), and **(d)** PhysicsDNA [15].

6.1 Real-world data

We use the physics model proposed by Jin et al. [10] (Equation (4)) within the physics-informed machine learning approaches (APHYNITY and PhysicsDNA). Table 1 shows the root mean square error (RMSE) for two test settings: **(a)** in-distribution, when the test data consists of unseen sensors with similar manufacturing conditions as in training, and **(b)** out-of-distribution, when the test data consists of unseen sensors manufactured under different conditions.

Training errors for NeuralODE and PhysicsDNA are much lower compared to APHYNITY since the latter does not use the historical measurements and is unable to fit the training data. Standalone physics model is also unable to obtain a perfect fit given only a few hours of initial voltage measurements. A similar trend is observed for in-distribution test errors computed over an unseen set of test sensors that were manufactured under the same conditions as the training sensors.

	Train RMSE	Test RMSE (mV)	
Method		In-distribution	Out-of-distribution
NeuralODE [2]	1.05	15.66	134.81
Physics model [10]	39.69	42.05	110.94
APHYNITY [8]	35.17	40.84	108.55
PhysicsDNA	1.36	15.10	**62.30**

Table 1: (Real-world data) Root mean squared error (mV) for training, in-distribution and out-of-distribution (OOD) test datasets.

The out-of-distribution test error of NeuralODE, a purely data-driven approach, is the highest. Since APHYNITY does not make use of historical observations, it performs similarly to the physics model. PhysicsDNA has much lower out-of-distribution error than the other methods ($\approx 1.75\times$ lower than the best competing method). However, the corresponding OOD test error is still much higher than its training error.

Our experiment shows that when the unseen test sensors have similar properties to the sensors in training, we can use a purely data-driven approach such as NeuralODE to solve the task. However, such an approach is not robust when the manufacturing conditions change for the new test sensors. The proposed approach performs similar to a data-driven approach over in-distribution examples, but is much more robust for out-of-distribution examples. Finally, we show that even state-of-the-art physics-informed machine learning methods (while better than standard ML methods) do not perfectly extrapolate to out-of-distribution data showcasing that radically new approaches are needed.

6.2 Synthetic data

As described in the Section 4.2, we generate the synthetic data using the physics model described in Equation (4) with different values for the sensor thickness parameter h. We use a simpler physics model (Equation (5)) within the physics-informed machine learning approaches (APHYNITY and PhysicsDNA).

Table 2 shows the root mean square error (RMSE) for two test settings: **(a)** in-distribution, when the test data consists of unseen sensors with thicknesses in the range

Method	Train RMSE	Test RMSE (mV)	
		In-distribution	Out-of-distribution
NeuralODE [2]	0.01	1.44	11.87
Physics model [10]	19.30	19.53	15.96
APHYNITY [8]	0.01	2.01	32.76
PhysicsDNA	0.01	2.80	**6.73**

Table 2: (Synthetic data) Root mean squared error (mV) for training, in-distribution test and out-of-distribution (OOD) test datasets.

50-60 microns, and **(b)** out-of-distribution, when the test data consists of unseen sensors with thicknesses in the range 150-160 microns.

All deep learning based models (NeuralODE, APHYNITY, PhysicsDNA) are able to fit the synthetic training data with very low errors. For the in-distribution test data, these methods perform comparably and outperform the physics model. This is because the standalone physics model is unable to obtain a perfect fit given only a few hours of initial voltage measurements. Out-of-distribution error of PhysicsDNA is much lower than the competing baselines: $\approx 1.7\times$ lower than NeuralODE and $\approx 5\times$ lower than APHYNITY.

Our experiment shows that any of the tested deep learning based method can be used when the test sensors are expected to be manufactured under similar conditions to those in training. However, PhysicsDNA is expected to be most robust for the out-of-distribution test sensors.

7 CONCLUSION

We explored the task of accelerated testing of roll-to-roll printed sensors using machine learning. We showed that due to shifts in manufacturing conditions, the prediction task requires models that are robust out-of-distribution.

Our real-world and synthetic experiments show that one can use purely data-driven methods to predict the future sensor behavior when the unseen sensors have similar manufacturing conditions as those observed in training. However, these methods are not expected to be robust for the out-of-distribution examples.

We described a physics-informed machine learning model, PhysicsDNA, that performs as well as purely data-driven methods for the in-distribution sensors, but is far more robust to out-of-distribution changes in the manufacturing conditions. While none of the tested methods perfectly extrapolate to the out-of-distribution examples, synthetic experiments suggest that PhysicsDNA is expected to perform well if the changes in manufacturing conditions are minimal.

ACKNOWLEDGMENTS

This work was funded by the Wabash Heartland Innovation Network and the SMART films consortium. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Steven L. Brunton, Joshua L. Proctor, J. Nathan Kutz, and William Bialek. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 113(15):3932–3937, 2016. ISSN 10916490. doi: 10.1073/pnas.1517384113.

[2] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 6572–6583, Red Hook, NY, USA, December 2018. Curran Associates Inc.

[3] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian Neural Networks. *arXiv:2003.04630 [physics, stat]*, July 2020.

[4] James H Faghmous and Vipin Kumar. A big data guide to understanding climate change: The case for theory-guided data science. *Big data*, 2(3):155–163, 2014.

[5] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups. *arXiv:2104.09459 [cs, math, stat]*, April 2021.

[6] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut Learning in Deep Neural Networks. *arXiv:2004.07780 [cs, q-bio]*, April 2020.

[7] Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian Neural Networks. *arXiv:1906.01563 [cs]*, September 2019.

[8] Vincent Le Guen, Yuan Yin, Jérémie Dona, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Thome, and Patrick Gallinari. Augmenting Physical Models

with Deep Networks for Complex Dynamics Forecasting. *arXiv:2010.04456 [cs, stat]*, October 2020.

[9] Chiyu "Max" Jiang, Karthik Kashinath, Prabhat, and Philip Marcus. Enforcing Physical Constraints in Neural Neural Networks through Differentiable PDE Layer. September 2019.

[10] Xin Jin, Ajanta Saha, Hongjie Jiang, Muhammed R Oduncu, Qingyu Yang, Sotoudeh Sedaghat, Darrel Kerry Maize, Jan P Allebach, Ali Shakouri, Nicholas J Glassmaker, Alexander Wei, Rahim Rahimi, and Muhammad Alam. Steady-State and Transient Performance of Ion-Sensitive Electrodes Suitable for Wearable and Implantable Electrochemical Sensing. *IEEE Transactions on Biomedical Engineering*, pages 1–1, 2021. ISSN 1558-2531. doi: 10.1109/TBME.2021.3087444.

[11] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (PGNN): An application in lake temperature modeling. 2017.

[12] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, November 2016. ISSN 0022-1120, 1469-7645. doi: 10.1017/jfm.2016.615.

[13] Georg Martius and Christoph H. Lampert. Extrapolation and learning equations. *arXiv:1610.02995 [cs]*, October 2016.

[14] Viraj Mehta, Ian Char, Willie Neiswanger, Youngseog Chung, Andrew Oakleigh Nelson, Mark D. Boyer, Egemen Kolemen, and Jeff Schneider. Neural Dynamical Systems: Balancing Structure and Flexibility in Physical Prediction. *arXiv:2006.12682 [cs, stat]*, April 2021.

[15] S Chandra Mouli and Bruno Ribeiro. Physics-first dynamic neural adaptation for more robust out-of-distribution predictions. *Manuscript in preparation*, 2022.

[16] Nikhil Muralidhar, Mohammad Raihanul Islam, Manish Marwah, Anuj Karpatne, and Naren Ramakrishnan. Incorporating Prior Domain Knowledge into Deep Neural Networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 36–45, December 2018. doi: 10.1109/BigData.2018.8621955.

[17] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. Technical report, 2018.

[18] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part II): Data-driven discovery of nonlinear partial differential equations. November 2017.

[19] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 2009.

[20] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards Physics-informed Deep Learning for Turbulent Flow Prediction. *arXiv:1911.08655 [physics, stat]*, June 2020.

[21] Rui Wang, Robin Walters, and Rose Yu. Incorporating Symmetry into Deep Dynamics Models for Improved Generalization. February 2020.

[22] Alireza Yazdani, Lu Lu, Maziar Raissi, and George Em Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLoS computational biology*, 16(11): e1007575, 2020.