PI-LSTM: Physics-Infused Long Short-Term Memory Network

1st Shubhendu Kumar Singh
Mechanical and Aersopace Engineering
University at Buffalo
Buffalo, USA
ssingh55@buffalo.edu

2nd Ruoyu Yang
Mechanical and Aersopace Engineering
University at Buffalo
Buffalo, USA
ruoyuyan@buffalo.edu

3rd Amir Behjat
Mechanical and Aersopace Engineering
University at Buffalo
Buffalo, USA
amirbehj@buffalo.edu

4th Rahul Rai
Mechanical and Aersopace Engineering
University at Buffalo
Buffalo, USA
rahulrai@buffalo.edu

5th Souma Chowdhury
Mechanical and Aersopace Engineering
University at Buffalo
Buffalo, USA
soumacho@buffalo.edu

6th Ion Matei
Palo Alto Research Center
Palo Alto, USA
imatei@parc.com

***Abstract*—We introduce a novel machine learning based fusion model, termed as PI-LSTM (Physics-Infused Long Short-Term Memory Networks) that integrates first principle Physics-Based Models and Long Short-Term Memory (LSTM) network. Our architecture aims at combining equation-based models with data-driven machine learning models to enable accurate predictions of complex dynamic systems. In this hybrid architecture, recurrency aids the temporal memory of the inputs and output of the partial physics model, in a way that facilitates generalization with scarce data sets. We illustrate the application of PI-LSTM on two dynamical systems namely Inverted Pendulum and Tumor Growth. Empirical results on both test problems stand witness to the effectiveness of using physics in guiding machine learning models and the superiority of the outlined hybrid model over purely data-driven models.**

***Index Terms*—Hybrid Models, Long Short-Term Memory, Physics-Based Model, Hybrid Modeling Metrics and Standard Problems.**

I. Introduction

Predicting complex physical system behavior is important for many science and engineering applications. A multitude of physical systems applications including design, control, diagnosis, and prognostics are predicated on the assumption of model availability. There are mainly two approaches to modeling: Machine Learning (ML) and Physics-based modeling (Model-Based, MB). Both of these approaches have their limitations in most real world applications (discussed next).

Typical data-driven ML methods create and train black-box models, which are prone to a number of limitations. These models fail to imbibe the underlying physical principles that govern the real-world phenomena. Additionally, data-driven models require large amounts of labeled data-sets for achieving higher accuracies, which is often unavailable in real-world engineering problems. Further, their performance depends on the quality of available data. When the training sets are small the performance of these black-box models exhibit poor performance and they fail to generalize well beyond seen environments/data [1]. Another significant drawback of data-driven models is their complete disregard of the physical laws behind the data-sets. ML models turn blind eyes towards the essential laws of physics and thus lack consistency in results that could be used reliably to model physical systems. Even if the performance prediction of ML models is high, they lack the physical interpretability of the obtained output.

In contrast, Physics-based models (Model Based, MB), grounded in the core doctrines of science showcase explainable relationship between input and output variables [2], [3]. In the history of science and engineering, they have played a vital role in knowledge discovery in various technological domains. Although these models have solidified our understanding of the physical world, they are often difficult to build and heavily rely on expert knowledge. No model can exactly imitate the real physical processes they are supposed to encapsulate. In particular, this happens due to simplifying assumptions made during the development of these models. For example, many physics-based models used for simulation of dynamical systems often lump several physical parameters into one physical parameter to reduce the model complexity.

To overcome the limitations of both the approaches we argue, it is vital to develop novel hybrid methods that combine physics equation based models (if and when available) with data-driven machine learning models to enable predictive modeling of complex physical systems, especially in the presence of imperfect models and sparse and noisy data. In this paper, we introduce a novel machine learning based fusion model, termed as PI-LSTM (Physics-Infused Long Short-Term Memory Networks) that integrates first principle Physics-Based Models and Long Short-Term Memory (LSTM) network. In particular, the main contributions of this paper are:

1) A novel hybrid model that integrate physics-based models with the recurrent memory structure of LSTM (Long

DARPA the Physics of Artificial Intelligence (**PAI**) program funding with Agreement No.: HR00111890037.

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00015

Short-Term Memory) networks to faithfully model dynamic systems.
2) It is shown that when it comes to hybrid models all physics models are not created equal. In other words, the performance of the hybrid models is dependent on the quality of physics models available.
3) Several metrics and couple of new problems specifically suited to hybrid modeling domain have been outlined. The outlined metrics and problems could potentially serve as benchmarks for researcher focusing on hybrid modeling problems.

The *hybrid* modeling approach has been applied to two different, yet illustrative problems: the Inverted Pendulum and the Cancer Cell Growth. Due to the dynamic nature of these two problems purely data-driven models require higher number of training samples to make accurate predictions. We demonstrate that by using physics-based knowledge in constructing and guiding data-driven models, the resulting hybrid model can achieve better prediction accuracy, reduced complexity, as well as scientific interpretability of results with lower number of training samples.

Paper structure: We start with a related work survey, followed by the Hybrid model Description. We continue with the description of the example problems and comparison metrics. Performance comparison between hybrid models and data-driven models is outlined next. We end with a Discussion over the results and the future scope of this work.

II. Literature Survey

Our work draws motivation from the previous work where physics is integrated with machine learning models in order to enhance the predictive capabilities of the overall hybrid model [4]–[8]. In most of these works it is clearly evident that the physics of the system is pivotal in guiding the data-driven predictions. Although the works mentioned above focus on some aspect of hybrid modeling, they fail to exploit the temporal and spatial dependencies available in many problems and thus are vacuous or not applicable in the class of problems outlined in this paper.

Our hybrid model to a certain extent derives inspiration from the Long Short-Term Memory (LSTM) models [9], [10]. LSTM models clearly showcase the superior ability in capturing the dynamics of a system, thus making them front-runners during the selection process of a suitable ML architecture for the data-driven portion of our hybrid model. The memory cells in an LSTM unit helps to retain information from a certain instant in past which may at times prove to be significant in predicting the present and future states of a system [11], [12]. The unique properties of LSTM networks make them an ideal candidate for modeling problems with temporal and spatial dependencies that is pervasive in predictive analysis of complex physical systems.

III. Hybrid Model

In predictive modeling of complex physical systems, given the inputs X_{MB}, model parameters P_{MB} and the target variable Y_p the physics-based model *prescribes (not learn!)* a mapping $f_{MB} : [X_{MB}, P_{MB}] \rightarrow Y_p$. The data-driven machine learning approach, however, is focused on *learning* a model $f_{ML} : [X_{ML}] \rightarrow Y$ over a training set to produce estimates of output $\hat{Y}$ that are close to Y given inputs X_{ML}. As discussed before, both the approaches have inherent drawbacks.

In our hybrid model, we propose to integrate the sensor data or black box model output Y and output Y_p generated by the physics model as inputs to the LSTM model. Our hybrid model learns a mapping $f_{hybrid} : [X_{ML}, Y_p] \rightarrow$ Y. In the physics model, the output Y_p may be an insufficient representation of real phenomenon occurring in a dynamical system due to simplified or partial physics assumptions. By including Y_p in the hybrid model we fuse the physics-based equation information into the data-driven model. Thus, the hybrid model showcases better predictability and interpretability when compared to the purely data-driven model.

In LSTM based data-driven model module of our hybrid model, we integrate the sensor data and the physics model output in a creative way, by exploiting the recurrent processing power and memory retention capabilities of LSTM cells in a novel manner. Rather than just relying on time-series dependency of the datasets, like the conventional method, we set the number of time steps (or recurrent processings of each unit cell) equal to the number of features possessed by each sample data. Hence, we make our model not only limited to time-series or sequence-based data samples. For example, in the case of the Inverted Pendulum problem (described in next section), we have six input features and four outputs of the physics model. Thus, our hybrid model will have LSTM cells with ten (six plus four) time (or recurrent) steps each. On the other hand, purely data-driven (LSTM) model will only take six inputs directly from the sensor (black-box) and will have LSTM cells with six-time steps each. This approach ensures that we unleash the strong processing capabilities of the recurrent memory cells. Due to repetitive processing across successive time steps and information transfer from previous steps and data retention in memory cells, each feature gets more weight in determining the final output. The complete hybrid model is depicted in Fig. 1.

The working of each LSTM cell in our hybrid architecture can be described as follows. If x_n is one of the feature inputs at any time step t, the LSTM model generates hidden activations a^t, represented by (3), at each time step, which are further used for making predictions. The LSTM model defines a transition relationship for the hidden representation a^t through an LSTM cell, which takes the input x_n at the current time step and also the acquired information from previous steps. Thus when our LSTM network takes one feature as input, it is processed and the inherited information is passed on to the next step. Each LSTM cell contains a cell state c^t, calculated using (2), which serves as a memory and helps hidden units a^t in retaining information from the past. We generate a new candidate $c^{\sim t}$, using (1), to initially replace c^t as a placeholder. The cell state c^t is generated by combining c^{t-1}, a^{t-1}, and the input features at t.

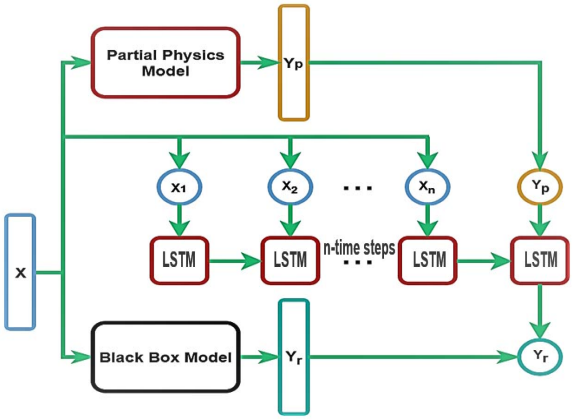


Fig. 1. Flow chart for LSTM based model

Algorithm 1 PI-LSTM

Input: $X_{r(mXn)}$
$\mathbf{Y}_{phy(mXk)} = Physics(\mathbf{X}_r)$
for $i = 1$ **to** n **do**
$\mathbf{I}_{mX(n+k)} = [\mathbf{X}_r, Y_{phy}]$
Model= LSTM()
Model.fit(I)
Predict= Model.predict(Test)
ERROR= MSE(Predict,Test label)
end for
$\text{ERROR}_{avg} = ERROR/n$

$$c^{\sim t} = tanh(W_a^c a^{t-1} + W_x^c x_n) \quad (1)$$
$$c^t = f^t \bigotimes c^{t-1} + u^t \bigotimes c^{\sim t} \quad (2)$$
$$a^t = o^t \bigotimes tanh(c^t). \quad (3)$$

Here $W_a^c \in R^{HxH}$ and $W_x^c \in R^{HxD}$ denote the weight parameters used to generate candidate cell state. Hereinafter we omit the bias terms as they can be absorbed into weight matrices. Then, we generate a forget gate layer f^t, an update gate layer u^t, and an output gate layer, as:

$$f^t = \sigma(W_a^f a^{t-1} + W_x^f x_n) \quad (4)$$
$$u^t = \sigma(W_a^u a^{t-1} + W_x^u x_n) \quad (5)$$
$$o^t = \sigma(W_a^o a^{t-1} + W_x^o x_n) \quad (6)$$

At any step t, the hidden representation a^t is an accumulation of information from previously processed features and hence affects the generation of final output. In the subsequent time step $t+1$, the next feature Y_p of the input sample, which is physics model output, is fed to the model along with the information a^t, from the previous step. Now we calculate the attributes of this time steps using (7) to (9).

$$c^{\sim t+1} = tanh(W_a^c a^t + W_x^c Y_p) \quad (7)$$
$$c^{t+1} = f^{t+1} \bigotimes c^t + u^{t+1} \bigotimes c^{\sim t+1} \quad (8)$$
$$a^{t+1} = o^{t+1} \bigotimes tanh(c^{t+1}). \quad (9)$$

The transition of cell state over time forms a memory flow that enables the modeling of long-term spatial and temporal dependencies. All the input features from sensor sample data and features of a sample from the physics model go through the aforementioned processing sequence.

IV. Problem Description

In this section, we give a brief description of the example problems considered in this paper and also introduce the mathematical descriptions of the physics-based models used in the example problems.

A. Inverted Pendulum

The inverted-pendulum system (IPS) is one of the most popular dynamical systems examples of an unstable system being used in machine learning domain (e.g., reinforcement learning) or controls. In IPS problem, the angle and angular velocity of a pendulum are controlled by moving the cart Fig. 2. The position of the cart and the force applied to it are bounded.

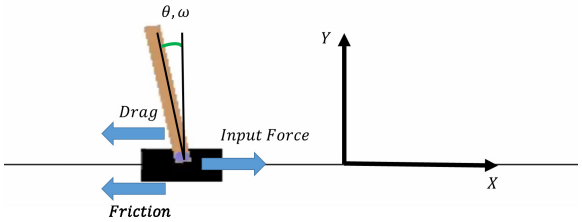


Fig. 2. Inverted Pendulum Problem

Two types of *modified IPS* partial physics models are considered while formulating the problem statement. In both models, average wind speed is considered, which differs from the real wind speed. The only difference between these two models is the coefficient of friction involved.

Fusion of two levels of physics, with our data-driven LSTM network, yields two types of hybrid models namely Hybrid 1 and Hybrid 2. In a later section of this paper it is shown that the quality of physics model or its abstraction ability in its closeness to the real system behavior, will ultimately help in improving the prediction accuracy of the overall hybrid model. The hybrid model aims to determine the state of the IPS with respect to different spatial and temporal coordinates. The state vector includes X, V, θ, ω, which are the position, the velocity of the cart, the angle and the angular velocity of the pendulum, respectively. The state space equations used for both the partial and the complete physics models are:

$$\begin{bmatrix} \dot{X} \\ \dot{V} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} V \\ \frac{F+M_p sin(\theta)(l\dot{\theta}^2 - gcos(\theta))}{M_c + M_p sin(\theta)^2} \\ \omega \\ \frac{-Fcos(\theta) - M_p l\dot{\theta}^2 sin(\theta)cos(\theta) + (M_p+M_c)gsin(\theta)}{l(M_c + M_p sin(\theta)^2)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.001 \end{bmatrix} \times F \quad (10)$$

where M_c, M_P, l, g are the mass of cart, the mass of the pole, the length of the pole and the gravitational acceleration,

respectively. We denote the applied force by F. To make the model more realistic, this term is modified considering *air drag* and *friction*:

$$\tilde{F} = F - F_D - F_f \tag{11}$$

$$F_D = \frac{\rho_{Air} C_D A_C (V + V_{wind} cos(\phi_{wind}))^2}{2} \tag{12}$$

$$F_f = \mu(M_c + M_P)g \tag{13}$$

where F_D and F_f denote the air drag and friction forces respectively. The two additional forces, change the force $\tilde{F}$ applied to the cart and are considered for both partial and complete physics models. The partial physics model uses average values for both the wind and friction ((14)), while in the detailed model both of these attributes are variable ((15)).

$$\textit{partial physics} = \begin{bmatrix} \mathrm{V}_w(t) \\ \mu(t) \end{bmatrix} = \begin{bmatrix} \mathrm{V}_{w,0} \\ \mu_0 \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} \mathrm{V}_w(t) \\ \mu(t) \end{bmatrix} = \begin{bmatrix} \mathrm{V}_{w,0} + \sum_{i=1}^{4} A_{w,i} sin(\omega_{w,i} + \phi_{w,i}) \\ \mu_0 + \sum_{i=1}^{4} A_{\mu,i} sin(\omega_{\mu,i} + \phi_{\mu,i}) \end{bmatrix} \tag{15}$$

The input to our hybrid model contains initial states, input force and time along with the four output states of the physics model, making the total input parameter equal to 10. The outputs (to be predicted by the hybrid model) are the four states that the system reaches after applying the input force.

$$f(\begin{bmatrix} \mathrm{X}_0 \\ \mathrm{V}_0 \\ \theta_0 \\ \omega_0 \end{bmatrix}, F, t_f) = \begin{bmatrix} \mathrm{X}_f \\ \mathrm{V}_f \\ \theta_f \\ \omega_f \end{bmatrix} \tag{16}$$

Table I lists the parameter values used for different models of the Inverted Pendulum problem.

TABLE I
INVERTED PENDULUM PROBLEM: COMPLETE (REAL) AND PARTIAL PHYSICS MODELS

Model	Wind Speed		Friction
Complete Model	$V_{wind}(t)$	$=$	$\mu_k = 0.3$
	$4 + 0.4\sin(2.4t)$	$+$	
	$0.32\sin(5.4t)$	$+$	
	$1.6\sin(0.2t + 0.2)$	$+$	
	$0.4\sin(176t - 0.04)$		
Partial Model - I	$V_{wind}(t) = 4$		$\mu_k = 0.3$
Partial Model - II	$V_{wind}(t) = 4$		$\mu_k = 0.03$

B. Cancer Cell Growth

Cancer cells exhibit uncontrolled growth and no longer respond to neural signals that control cell growth inside the body. Cancer cells originate within the tissues and keep on dividing. In the later stages, these cells metastasize, meaning they will break through tissues and spread to nearby tissues and other organs. Our objective is to model the growth of cancer cells, i.e., to estimate the size of the tumor based on its initial size and growth rate of cells.

The physics models characterizing tumor growth can be classified into two classes: (i) closed-form differential equation-based models [13]–[16] and (ii) discrete models based on cellular automata [17], [18] and agent-based models [19]. Discrete models are closer to reality when compared to mathematical equation-based models, but are computationally expensive. We used the later to generate training data. As a partial physics-based model, we use one based on Ordinary Differential Equations (ODEs). ODEs based partial physics model can estimate tumor growth, based on differential equations, but with less accuracy. The hybrid model proposed in this paper will use this partial physics model to predict tumor cell growth.

The cellular automata model classifies the cells into three categories: Stem cells, semi-differentiated cells, and fully differentiated cells. Their cellular differentiation mechanism is depicted in Fig. 3. Only the stem cell has the ability to divide infinitely while the other two types have limited divisibility. Stem cells can divide symmetrically to produce new stem cells known as daughter cells, and hence can increase the tumor size. Also, through asymmetric division, they can produce differentiated cells, which can only divide to a certain extent. Semi-differentiated cells can divide up to some extent, while the fully differentiated ones cannot divide at all. Therefore the probability to divide for each cell, p_d, and the probability of having symmetric or asymmetric division, p_{as} are considered as parameters for the model. Differentiated cells die due to factors such as lack of glucose or attack from immune cells. Thus, in the cellular automata model, the rate of cell death α is one of the important parameters. In this model, the overall running time is also one of the input parameters.

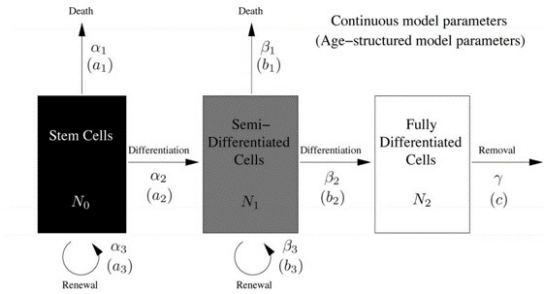


Fig. 3. Cell differentiation [20]

Our physics model traces its roots to a model proposed by [20]. In this model, a non-linear ODE is used to model tumor growth and stability criterion for the system. This model is generalized by considering 12 states, two of the states being the number of stem and fully-differentiated cells. Semi-differentiated cells can have 10 divisions before acquiring fully-differentiated state. The rest of the ten states are characterized by the number of cells corresponding to each of these 10 divisible cellular conditions. These states are governed by (17) to (21).

$$X = [N_s, N_{d,10}, N_{d,9}, ..., N_{d,1}, N_{d,0}]^T \tag{17}$$

$$\frac{dN_s}{dt} = r_s N_s \tag{18}$$

$$\frac{dN_{d,10}}{dt} = r_{as} N_s - \alpha N_{d,10} - r_d N_{d,10} \tag{19}$$

$$\frac{dN_{d,i}}{dt} = 2r_d N_{d,i+1} - \alpha N_{d,i} - r_d N_{d,i}, 1 \leq i \leq 9 \tag{20}$$

$$\frac{dN_{d,0}}{dt} = 2r_d N_{d,1} - \alpha N_{d,0} \tag{21}$$

where, X represents the state vector, N_s is the Number of stem cells, $N_{d,i}$ is the number of differentiated cells with i remaining divisions, $N_{d,0}$ is the number of fully differentiated cells, and α is the removal rate for differentiated cells and r_d is the division rate of these cells. Each division generates two cells. For Stem cell, the rates of symmetric and asymmetric division are r_s and r_{as} respectively.

Despite being an attractive approach, estimating the number of cells, at different levels of divisibility potential, is very challenging. Therefore another model is proposed which is also the physics model used for our hybrid architecture. This model, instead of calculating each cell type, considers the total population of cells. Equation (22) explains this new input.

$$N = N_s + \sum_{i=0}^{10} N_{d,i} \tag{22}$$

The total input in this model is five, the total cell population N being one of the inputs while the remaining four inputs are the parameters p_d, p_{as}, α, and the time difference Δt. The physics-based model used in our hybrid model has one output, i.e., the total cell population after time Δt. In the hybrid model, the LSTM network takes a total of six inputs, five being the same as those of the physics-based model and the remaining one is the physics model's output itself.

V. Performance Metrics

We outline a set of performance metrics that enables one to compare the performance of hybrid models and purely data-driven approaches by assessing the predictive capabilities of both the models as well as their ability to capture dynamic characteristics of the system. Next, five different metrics are described in details.

A. Generalizability

It refers to the ability of a model to perform well on unseen data sets within its training input domain range. In this paper, we use MSE (Mean Squared Error) as the evaluation criteria for generalizability. Mean Square Error measures the average squared difference between the prediction value and the original truth value. The MSE can be expressed by (23) :

$$MSE = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n} \tag{23}$$

where $\hat{y}$ is the predicted value, y is the ground truth value and n is sample size.

B. Extrapolability

Extrapolability is the measure of a model's predictive accuracy on a data set outside its training range. Here, also we use MSE as the metric for evaluation. The difference as compared to the generalizability metric lies in the fact that generalizability focuses on interpolation whereas extrapolability focuses on extrapolation.

C. Model Complexity

This metric provides an account of the computational complexity of the given model. Based on the required number of flops for generating the output of architecture, the computation time is calculated. Memory is calculated by estimating the storage required for various model attributes like weights. Metric encapsulating computational time and memory requirement help in the trade-off between performance and computational speed.

D. Robustness to noisy input

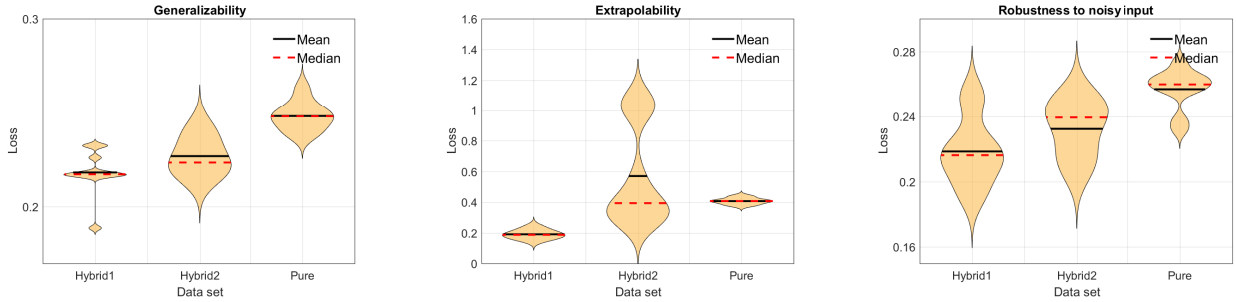
This metric is used to capture the stability of models in the presence of noisy data. Higher robustness of a model indicates its capability to perform even when accurate measurements are not available causing the measured readings to deviate from their real values. A robust model is useful in modeling real-world systems where noisy measurements are the norm.

E. Sensitivity to data set size

We use data-sets of different sizes to train both the data-driven and hybrid models. Observations are made to demonstrate the change in accuracy with varying data size. For comparison purposes, training algorithms hyperparameters such as learning rate, initial state, etc. are kept constant. The sensitivity is quantified using generalizability. Here also MSE is used as a metric for evaluation.

VI. Computational Performance Comparison

We demonstrate the ability of the proposed PI-LSTM hybrid model to perform better in the presence of the abstracted physics model when compared to purely data-driven approach (LSTM model). We use data generated from the detailed model as a black box to train the purely data-driven LSTM and coarse physics-based models to generate simulated data that is fed into PI-LSTM hybrid model whose data-driven module is trained with data from the detailed model. Different Hyperparameter combinations(such as learning rate, epochs, layers, etc.) were tried to select the best of models. Fig. 5 showcases a few samples from the hyperparameter tuning process for hybrid 1 model. The box plots are generated by running 5 iterations of each combination. Both the hybrid and data-driven model comprise of an LSTM layer and a dense output layer. LSTM layer has 20 memory units. The stipulated time steps per unit vary depending on the number of input features as mentioned previously. In the case of Inverted Pendulum example, the output layer has four unit cells, while for the cancer growth it has one output unit cell. We investigated the performance of the both models on GPU (NVIDIA GTX 1070) with Intel Core i7-8750H @2.20 GHz, 2208 MHz, 6 cores with 12 logical processors. The above-mentioned performance metrics are used for performance comparison purposes. The violin plots (in Figures such as Fig. 4(a)) are generated by running 10 epochs. Additional performance plots(Fig. 6, Fig. 7, Fig. 8), attributed to superior performance of hybrid model, are also included. *The code and dataset for the two problems can be found on github-https://github.com/shubhsingh55/code*. Next, we detail the results on two described example problems.



(a) Generalizability (b) Extrapolability (c) Robustness to noisy input

Fig. 4. Violin plot for generalizability, extrapolability, and robustness of inverted pendulum

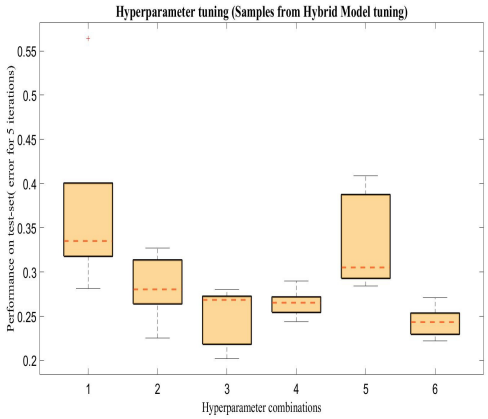


Fig. 5. Inverted Pendulum-Hyperparameter Tuning of Hybrid Model

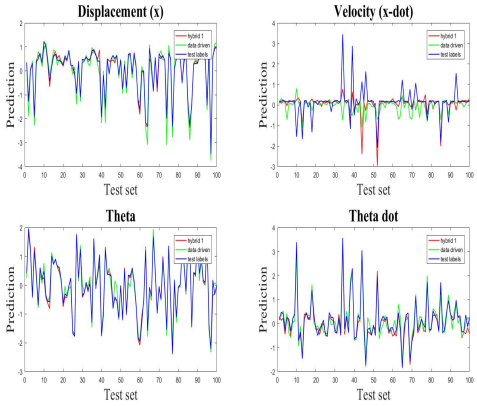


Fig. 8. Inverted Pendulum-Performance comparison over test set

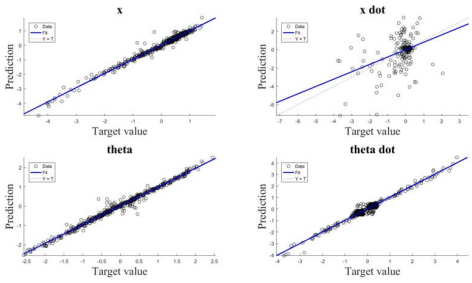


Fig. 6. Inverted Pendulum-Regression plot of Hybrid1 Model

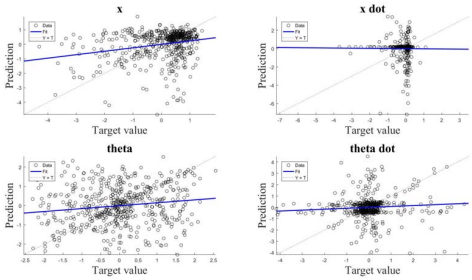


Fig. 7. Inverted Pendulum-Regression plot of Data-driven Model

A. *Inverted Pendulum*

1) Generalizability
Fig. 4(a) illustrates that the hybrid model has a low MSE range when compared to the data-driven model over the test set. This further showcases the better predictive capability of the hybrid model. The level 1 hybrid model is more accurate than the level 2 hybrid model. This illustrates that the quality of physics model (level of abstraction) affects the performance of the hybrid model.

2) Extrapolability
On dataset outside the training range, the range of MSE and median values for level 1 hybrid model is less than that of level 2 hybrid and data-driven models as shown in Fig. 4(b). In fact, the performance of data-driven model is better than that of the level 2 hybrid model. This illustrates that only physics models with sufficient abstraction of real problem are useful in hybrid modeling framework. *Although physics models provide supplementary information, poor quality physics model can lead to the poorer performance of hybrid model when compared to a purely data-driven model.*

3) Complexity
From the results depicted in Table II, it is evident

that the hybrid model does not compromise much on computational efficiency and memory with performance. Also, it can be observed that although hybrid 1 and hybrid 2 models have the same complexity, but due to better quality physics involved, hybrid 1 has a better performance to cost ratio than hybrid 2.

TABLE II
COMPLEXITY

Model	Flops	Total calculation	Memory
hybrid 1,hybrid 2	8918	4418	3208
pure data driven	5510	2810	2005

4) Robustness to noisy input
As depicted by Fig. 4(c), the hybrid models show better stability and perform better when compared to the data-driven model in the presence of noise. For this metric, the level 1 hybrid model outperforms the level 2 hybrid model.
5) Sensitivity to the number of samples
Increase in the size of training sample drastically improves the performance of hybrid models when compared to the data-driven model (Fig. 9). This can be attributed to better feature capturing ability of the hybrid model. As depicted in Fig. 9, the performance of level 1 hybrid model is better than that of level 2 hybrid model.

B. Cancer Cell Growth

1) Generalizability
As evident from Fig. 10(a), hybrid model records around 20 percent reduction in MSE values, as compared to pure data-driven model on the test set within the training range.
2) Extrapolability
On a dataset outside the training range, the median value for hybrid model is less than that of data-driven model (Fig. 10(b)), standing witness to hybrid model's better predictive capabilities even with unseen extrapolated data.
3) Complexity
From the results tabulated in Table III, it is evident that the hybrid model has a better trade-off between performance and computational cost when compared to a purely data-driven model.

TABLE III
COMPLEXITY

Model	Flops	Total calculation	Memory
hybrid	5210	2510	605
pure data driven	4358	2108	554

4) Robustness to noisy input
As depicted by Fig. 10(c), the hybrid model is superior in tackling noise when compared to the data-driven model.
5) Sensitivity to the number of samples
The response of the hybrid model to increasing data size is better than a purely data-driven model (Fig. 11). This illustrates that the hybrid model has better learning ability and eventually performs better with increasing data set size.

VII. Conclusion and Future Work

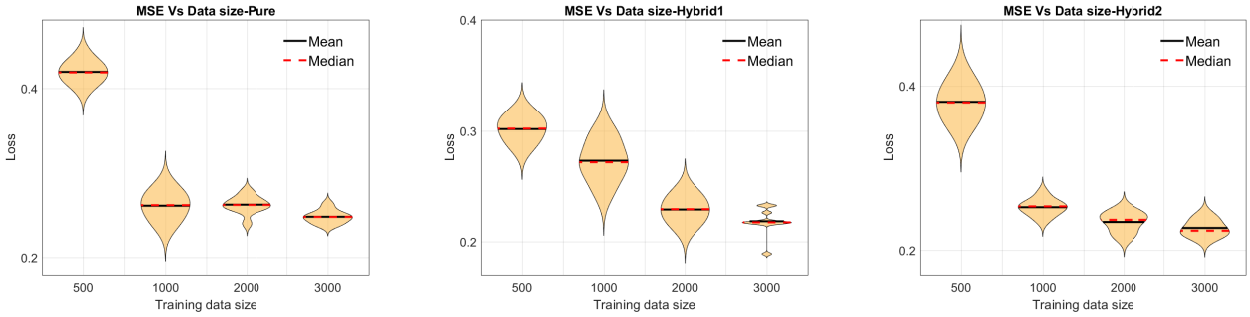
In this paper, a hybrid architecture that utilized a physics-based model for guiding the LSTM based machine learning model is outlined. The outlined hybrid model demonstrates the feasibility of integrating a physics-based model with ML models to model a physical system's dynamics. As evident from empirically derived results using different metrics, the performance of hybrid models when compared to purely data-driven model is better, for both the inverted pendulum and tumor growth problems.

From the results, it is empirically evident that the quality of the available physics model also plays a significant role in determining the performance of the hybrid models. Due to the better quality of physics involved, level 1 hybrid architecture delivers the best performance. On the other hand due to the involvement of comparatively lower quality physics, level 2 hybrid lags behind level 1 hybrid model in terms of performance. Thus, it can be emphasized that along with the efficiency of involved data-driven architectures, the quality of the physics-based models also affects the accuracy of the overall hybrid model.

An avenue for future work is developing different novel hybrid architectures based on different types of neural network architectures such as Generative adversarial networks (GANs) and neuro-evolution models. Creating a standardized set of problems that can serve as a benchmark for hybrid models is also a right direction for future work.

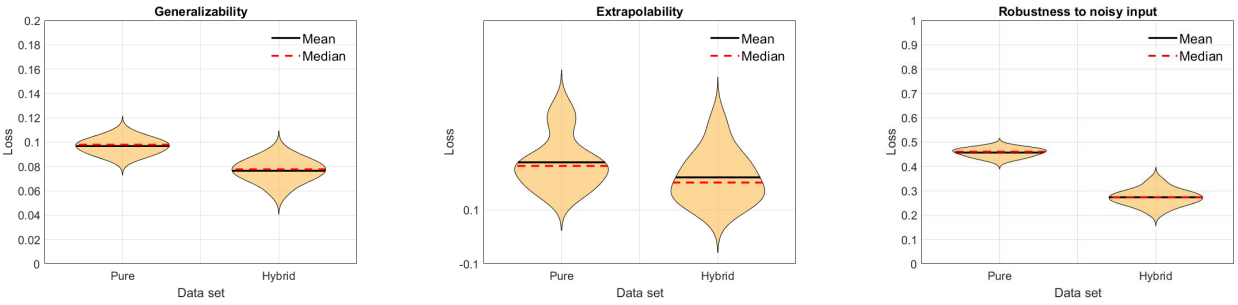
References

[1] Jonathan, T., Gerald, A., et al," Special issue: dealing withdata.Science," 331(6018):639-806, 2011.
[2] Araujo, R. P. and McElwain, D. S, " A history of the study of solid tumour growth: the contribution of mathematical modelling," Bulletin of mathematical biology, 66(5):10391091, 2004.
[3] Bresch, D., Colin, T., Grenier, E., Ribba, B., and Saut,O, "Computational modeling of solid tumor growth: the avascular stage," SIAM Journal on Scientific Computing,32(4):2321-2344, 2010.
[4] Hanachi, H., Yu, W., Kim, I., and Mechefske., C, "Hybrid physics-based and data-driven phm," 2017.
[5] Karpatne, A., Watkins, W., Read, J., and Kumar, V, "Physics-guided neural networks (pgnn): An application in laketemperature modeling," arXiv preprint arXiv:1710.11431,2017.
[6] Kristjanpoller, W., Fadic, A., and Minutolo, M. C, "Volatility forecast using hybrid neural network models," Expert Systems with Applications, 41(5):2437-2442, 2014.
[7] Pillai, P., Kaushik, A., Bhavikatti, S., Roy, A., and Kumar,V, "A hybrid approach for fusing physics and data for failure prediction," International Journal of Prognostics and Health Management, 7(025):1-12, 2016.
[8] Young, C.-C., Liu, W.-C., and Wu, M.-C, "A physically based and machine learning hybrid approach for accurate rainfall-runoff modeling during extreme typhoon events," Applied Soft Computing, 53:205-216, 2017.
[9] Hirose, N. and Tajima, R," Modeling of rolling friction byrecurrent neural network using lstm," In Robotics andAutomation (ICRA), 2017 IEEE International Conference on, pp. 6471-6478. IEEE, 2017.
[10] Mohajerin, N. and Waslander, S. L, "Multi-step prediction of dynamic systems with recurrent neural networks," arXivpreprint arXiv:1806.00526, 2018.



(a) Pure (b) Hybrid 1 (c) Hybrid 2

Fig. 9. Violin plot for sensitivity to the number of sample of inverted pendulum



(a) Generalizability (b) Extrapolability (c) Robustness to noisy input

Fig. 10. Violin plot for generalizability, extrapolability, and robustness of cancer cell growth model

(a) Pure (b) Hybrid

Fig. 11. Violin plot for sensitivity to the number of sample of cancer cell growth model

[11] Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury,J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R, "Ask me anything: Dynamic memory networks for natural language processing," In International Conference on Machine Learning, pp. 1378-1387, 2016.

[12] Swafford, Z. and Barron, A, "Lstms and dynamic memory networks for human-written simple question answering," Stanford, 2016.

[13] Murphy, H., Jaafari, H., and Dobrovolny, H. M, " Differ-ences in predictions of ode models of tumor growth: acautionary example,"BMC cancer, 16(1):163, 2016.

[14] Ribba, B., Holford, N. H., Magni, P., Troc oniz, I., Gue-orguieva, I., Girard, P., Sarr, C., Elishmereni, M., Kloft,C., and Friberg, L. E, " A review of mixed-effects models of tumor growth and effects of anticancer drug treatmentused in population analysis," CPT: pharmacometrics systems pharmacology, 3(5):110, 2014.

[15] Roose, T., Chapman, S. J., and Maini, P. K, " Mathematical models of avascular tumor growth," SIAM review, 49(2):179-208, 2007.

[16] Sachs, R., Hlatky, L., and Hahnfeldt, P, " Simple ode models of tumor growth and anti-angiogenic or radiation treat-ment," Mathematical and Computer Modelling, 33(12-13):1297-1305, 2001.

[17] Moreira, J. and Deutsch, A, " Cellular automaton models of tumor development: a critical review," Advances inC omplex Systems, 5(02n03):247-267, 2002.

[18] Poleszczuk, J. and Enderling, H, " A high-performance cellu-lar automaton model of tumor growth with dynamically growing domains," Applied mathematics, 5(1):144, 2014.

[19] Wang, Z., Butner, J. D., Kerketta, R., Cristini, V., and Deis-boeck, T. S, " Simulating cancer growth with multiscale agent-based modeling," In Seminars in cancer biology,volume 30, pp. 7078. Elsevier, 2015.

[20] Johnston, M. D., Edwards, C. M., Bodmer, W. F., Maini,P. K., and Chapman, S. J, "Mathematical modeling of cell population dynamics in the colonic crypt and in colorectalcancer," Proceedings of the National Academy of Sciences,104(10):4008-4013, 2007.