



Newtonian Physics Informed Neural Network (NwPiNN) for Spatio-Temporal Forecast of Visual Data

Anurag Dutta¹ · K. Lakshmanan² · Sanjeev Kumar³ · A. Ramamoorthy⁴

Received: 19 February 2024 / Accepted: 10 April 2024
© The Author(s) 2024

Abstract

Machine intelligence is at great height these days and has been evident with its effective provenance in almost all domains of science and technology. This work will focus on one handy and profound application of machine intelligence-time series forecast, and that too on visual data points, i.e., our objective is to predict future visual data points, given a subtle lag to work on. For the same, we would propose a deep learner, Newtonian physics informed neural network (NwPiNN) with the critical modelling capabilities of the physics informed neural networks, modelled on the laws of Newtonian physics. For computational efficacy, we would work on the gray-scale values of pixels. Since the variation in data pixel values is not only provoked by the pixel gray values but also by the velocity component of each pixel, the final prediction of the model would be a weighted average of the gray value forecast and the kinematics of each pixel, as modelled by the PiNN. Besides its' proposal, NwPiNN is subjected to benchmark visual dataset, and compared with existing models for visual time series forecast, like ConvLSTM, and CNN-LSTM, and in most of the occasions, NwPiNN is found to outperform its preliminaries.

Keywords Visual data · Physics informed neural networks · Spatio-temporal analysis · Time series forecast

1 Introduction

Time series is a statistical term for a collection of non-static data points [10] with temporal coherence between them. Time series forecasting involves predicting futuristic data

points in time by statistical or intellectual means. Traditionally, time series prediction finds its intuition from statistical inference methodologies, for example, autoregressive integrated moving average (ARIMA) [23], simple exponential smoothing (SES) [14], etc. In recent times, machine intelligence has been notable, and they are being used in almost every domain of science and technology, be it, engineering [29], medicine [28], or even the legislature [31]. Thereby, machine learning is also utilized for time series forecasting, for example, long short-term memory (LSTM) [15], reservoir computing (RCN) [32], etc.

One of the best exemplary uses of time series forecasting was evident during the COVID-19 Pandemic; several researchers, across the statistical and computer domains came forward to provide defense against the virus. This resulted in the development of novel epidemic forecasting models, epidemiologically-informed neural networks (EINNs) [30], epidemiological priors informed deep neural networks (Epi-DNNs) [25], etc. Post-COVID-19 saw a boom in Machine Intelligence, be it, large language models (LLMs) [8], natural language processing (NLP) [16], generative adversarial networks (GANs) [6], etc.

Mostly, time series is used for forecasting numeric data points, but with this, in our research, our target is visual data

✉ Anurag Dutta
anuragdutta@ieee.org

K. Lakshmanan
lakshmanlingam@kas.edu.kw

Sanjeev Kumar
Sanjeevnm83@gmail.com

A. Ramamoorthy
ramamoorthy@velammal.edu.in

¹ Department of Computer Science and Engineering, Government College of Engineering and Textile Technology, Calcutta, India

² Department of Mathematics, Kuwait American School of Education, Salmiya, Hawalli, Kuwait

³ Department of Computer Science, G. L. Bajaj Institute of Technology and Management, Greater Noida, Uttar Pradesh, India

⁴ Department of Mathematics, Velammal Engineering College, Chennai, Tamil Nadu, India

points, (or multimedia). Multimedia is a stream of binary digits [17], with added semantics. For this research, we are considering the most important component of multimedia-images. Our objective is to predict the next Frame, from a set of previous frames as a lag. Recent works on time series forecast focus on two important aspects of forecast, *length of forecast*, and *computational time of forecast*.

Initially, LSTMs were used for the forecast. Still, it was difficult to scale up using LSTM units, later on with the innovation of attention-based strategies [34], Transformers came up with their stack-based architecture which is easy to scale. But the main aspects were still lagging. They were not computationally efficient. Then came, the lightweight stack-block type architecture [36], Neural basis expansion analysis for interpretable time series forecasting (NBEATS) [27], neural basis expansion analysis with exogenous variables (NBEATSx) [26], neural hierarchical interpolation for time series forecasting (NHITS) [7], etc. Alongside these, the trivial models came up again hybridizing with loss functions inspired by physical differentials like the works of Elabid et al. [13], and Dutta et al. [12], performing almost similarly to the stack-block type architecture. While the former focused on a unit-step-ahead forecast, the latter focused on extending the paradigm to a multi-step-ahead forecast. Besides, the latest work by Cerqueira et al. [5] contributed extensively to the existing literature by proposing, a novel methodology, forecasted trajectory neighbors (FTN), that claims to be integrated with existing multi-step-ahead forecasting algorithms provoking substantial improvements in their performance. Drawing from the positives of these paradigms, we will propose, *newtonian physics informed neural network* (NwPiNN), a physics informed neural network built on the laws of Newtonian (or classical) physics, thus giving it the ability to forecast alongside modelling the perturbations in the forecasting lag.

The reason behind the choice of a PINN for the forecast is that the forecast of pixel gray values (which is the main objective of the research) will not only be dependent on the base pixel gray values throughout the lag but also, it will be affected by the velocity and acceleration possessed by pixels. If it had been simply the base gray values, lightweight models like NBEATS, or NHITS would have taken the pie, but since it involves the perturbations of the velocity and acceleration of each pixel's values, a physics informed neural network [4] is needed to model its parameters accordingly.

Newtonian Laws of Motion is an established set of conceptualizations that succeeds in explaining the dynamics of several (non-chaotic) observations that we witness around us. Since the targeted input for this proposal is a sequence of frames, containing visual data; besides the proposal NwPiNN being a physics-informed neural network, for the modelling purpose, we adopt the Newtonian Laws for the physics behind the PINN. Further, suppose we peep deep into the

input characteristics, they are simply a sequence of pixels, with physical coherence between them, in terms of their derivatives of displacement, concerning the temporal dependence. For Example, if we break down a "One-Second Video" into say 24 frames, we can notice the coherence between consecutive frames [19]. Now, the relationship between pixels can be well visualized; some pixels will keep moving by a constant shift (displacement), while some will move with a constant scale (velocity, or higher order differentials of displacement), which can be well modelled by the PINN.

In recent times, much work is being done on the spatio-temporal forecast of the image sequence, For example, the work by Meyer, Langer et al. [21] on the forecast of fresh concrete properties concludes on the fact that their research can be modified by utilizing more of updated forecasting models like those based on Transformers, promising hope of improved performance. Another work by Arslan et al. [1] on the forecast of molecular profiling of multi-omic biomarkers using hematoxylin and eosin stained slide images comments on exploring assortment between tumour morphology, and model predictions, and this correlation is hoped to promise improved accuracy.

2 Problem Statement

Given a sequence of n frames of a $q \times p$ visual data, as n data points, $\mathcal{V}(n) = \{v_1, v_2, \dots, v_n\}$, we need to generate the spatio-temporal future data points, $v_{n+\lambda}$, $\forall \lambda > 0$, such that,

$$v_i = \begin{pmatrix} \gamma_{1,1,i} & \cdots & \gamma_{1,p,i} \\ \vdots & \ddots & \vdots \\ \gamma_{q,1,i} & \cdots & \gamma_{q,p,i} \end{pmatrix} \quad (1)$$

where, each $0 \leq \gamma_{x,y,i} \leq 2^8 - 1$ in (1) represents the gray value of the pixel in x th row, y th column, of the i th data point (or frame). For this research, we would consider, $\lambda = 1$. Therefore, the data stream is supposed to be cuboidal. Now, LSTM expects three-dimensional input in the form of (batch, seq_length, input_feature), to make the input data ($n \times p \times q$) compatible for the same, samples of seq_length n from each of $p \times q$ pixels were considered in iterations, with batch of 64 each with their respective input_feature.

3 Preliminaries

3.1 Long-Short Term Memory

The artificial neural network (ANN) [11], long short-term memory is utilized in machine learning and machine intelligence. LSTM features backpropagation instead of typical

feed-forward neural network [3] models. Such a neural network with recurrent connections may comprehend complete data streams and individual data pieces. Because of this feature, LSTM recurrent networks are perfect for handling and forecasting information. A cell, an input (I_t), an output gate (O_t), and a forget gate (F_t) make up a typical LSTM unit. The three gates determine how data moves through and out of the neuron, and the cell retains data across configurable time frames. By allocating a number between zero and one to a prior level in comparison to a current feed, forget gates choose which knowledge from a prior level to disregard. Considering there may be the latency of uncertain length amongst significant occurrences in a time series, LSTM recurrent networks are quite well suited to categorizing, interpreting, and predicting outcomes using time-series information. To solve the diminishing contour issue that can arise when learning conventional RNNs, LSTMs were created. The advantage of LSTM beyond RNNs concealed Markov models [9], and other succession learning approaches in many situations is their relative callousness to the separation distance. NwP-INN is also built on the realms of long short term memory. To adjust each component of the LSTM recurrent network following the gradient of the failure at the output nodes of the LSTM network, a Recurrent neural network [20] with LSTM blocks can be developed under supervision on a collection of training sequences using an optimizer-like gradient descent mixed with backpropagation throughout the duration to estimate the slopes desired throughout the optimization procedure.

3.2 Newtonian Laws of Kinematics

The movement of macroscopic objects, celestial objects, and galaxies, is described by the physical theory of classical mechanics. Sir Issac Newton is regarded as the father of classical physics (or Newtonian Physics) because of his immense contributions to the field of classical dynamics [24], be it Kinematics, Gravitation, or Optics, Newtonian Laws cover them all. For this research, we would stick to his contribution to Kinematics. Kinematics is a sub-branch of Classical Physics that defines the dynamics of particles, without considering the compelling force.

To begin with, we have the position vector, \mathbf{r} of any particle,

$$\mathbf{r} = x \hat{\mathbf{i}} + y \hat{\mathbf{j}} + z \hat{\mathbf{k}} \tag{2}$$

where, x, y, z are the coordinates, and $\hat{\mathbf{i}}, \hat{\mathbf{j}},$ and $\hat{\mathbf{k}}$ are the unit vectors along the respective axes.

The rate of change of the position vector (refer to Fig. 1), as in (2) with time, is velocity, \mathbf{v} ,

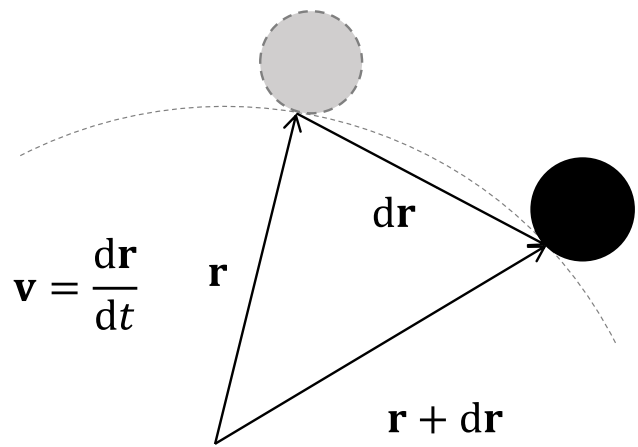


Fig. 1 Velocity of any particle, \mathbf{v} is computed by the gradient of the trajectory followed by the radial vector, \mathbf{r}

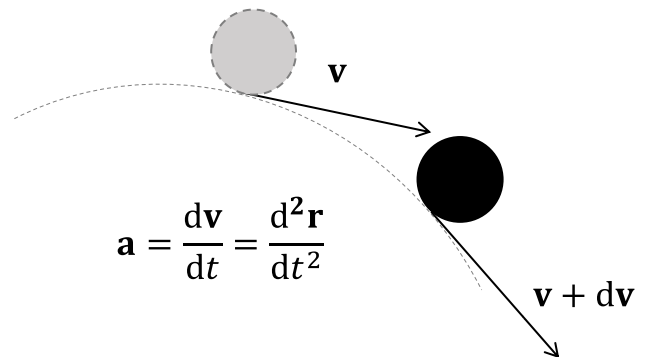


Fig. 2 Acceleration of any particle, \mathbf{a} is computed by the rate of change of velocity, \mathbf{v}

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = \left(\frac{dx}{dt}\right) \hat{\mathbf{i}} + \left(\frac{dy}{dt}\right) \hat{\mathbf{j}} + \left(\frac{dz}{dt}\right) \hat{\mathbf{k}} \tag{3}$$

Further, the rate of change of the velocity vector (refer to Fig. 2), as in (3) with time, is acceleration, \mathbf{a} ,

$$\mathbf{a} = \frac{d^2\mathbf{r}}{dt^2} = \left(\frac{d^2x}{dt^2}\right) \hat{\mathbf{i}} + \left(\frac{d^2y}{dt^2}\right) \hat{\mathbf{j}} + \left(\frac{d^2z}{dt^2}\right) \hat{\mathbf{k}} \tag{4}$$

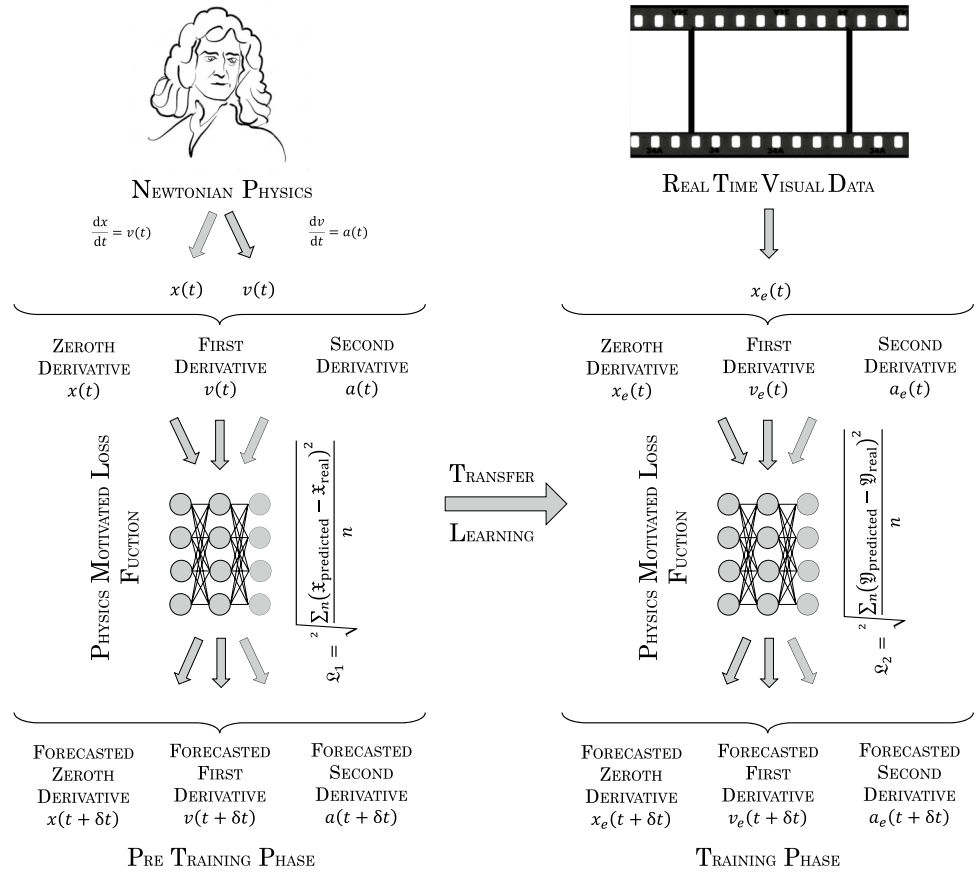
Kinematic equations relating (2), (3), and (4), termed as equations of motion are as follows,

$$|\mathbf{v}_t| = |\mathbf{v}_0| + |\mathbf{a}| \cdot t \tag{5}$$

$$|\mathbf{r}| = |\mathbf{v}_0| \cdot t + \frac{1}{2} \cdot |\mathbf{a}| \cdot t^2 \tag{6}$$

$$|\mathbf{v}_t|^2 - |\mathbf{v}_0|^2 = 2 \cdot |\mathbf{a}| \cdot |\mathbf{r}| \tag{7}$$

Fig. 3 Architectural view of the Newtonian physics informed neural network (NwPiNN). The model builds on the critical modelling capabilities of physics informed neural network to obtain the zeroth, first, second derivatives, x_{event} , v_{event} , and a_{event} , respectively. An LSTM unit is used as a back-end for NwPiNN, followed by a loss function, motivated by the laws of Newtonian physics for both the pre-training and training phases. The trained neuronal parameters (weights and biases) after ϵ_{max} epochs from the Pre-Training Phase are transferred to the training Phase. This procedure is followed for all the $q \times p$ pixels of the lag frames



Equations 5, 6, and 7 serve a pivotal role in Newtonian Physics Informed Neural Network (NwPiNN).

4 Newtonian Physics Informed Neural Network (NwPiNN)

This section lays the foundation, of Newtonian physics informed neural network (NwPiNN), building upon the single step ahead prediction framework of knowledge based deep learning (KDL) [13]. The proposal, unlike other physics informed neural networks, is free from chunks of parameters. The model is composed of 2 conjugated components, incentivized by transfer learning [33]. The first segment, pre-training of the recurrent neural network (RNN), is done by modelling the weights and biases using an artificial dataset, based on the dynamics of Newtonian mechanics, simulated using the Runge–Kutta method, to teach the exogenous harmonic perturbations in the neurons. This training (technically, pre-training) is done by incorporating a physical loss function based on Newtonian dynamics, supported by an adaptive moment estimation optimizer; because of this, the RNN will successfully capture disturbances in the historical

sequence across the temporal domain. The resolution of the non-linear equations is the *output*, and the temporal component is considered as an *input* in the conventional physics-informed neural network (PINN) paradigm. Differentiating the highly-grained feed-forward neural network and computing the temporal relationships through auto-differentiation are the steps involved hereby in figuring out the control parameter. Time-series data are discrete events without a robust mathematical model, and to control the variables over time is difficult, thus we create discrete time-series data components to overcome this challenge. The first order differentials, $\frac{d\phi}{d\psi}$ are computed using the First Order Principle [35] as follows,

$$\frac{d\phi}{d\psi} = \frac{\phi(\psi + \delta\psi) - \phi(\psi)}{\delta\psi}$$

The traditional physics-informed neural network paradigm backpropagates the temporal implications to adjust the network loss since the transfer function employed in multi-layered perceptron (MLP) [2] is differentiable. The network thereby forecasts (or computes), the values of the zeroth, first, and second order differentials, of the discrete series, $x(t)$.

The Loss function that initiates the backpropagation (both, for pre-training and training phases) for NwPiNN, is given in (8).

$$\mathfrak{L}_{\text{phy}} = \text{RMSE}(\mathfrak{Y}_{\text{pred}}, \mathfrak{Y}_{\text{real}}) \quad (8)$$

where,

$$\mathfrak{Y}_{\text{pred}} = 2 \cdot y_{\text{pred}} \cdot \frac{d^2 y_{\text{pred}}}{dt^2} - \left(\frac{dy_{\text{pred}}}{dt} \right)^2$$

and

$$\mathfrak{Y}_{\text{real}} = 2 \cdot y_{\text{real}} \cdot \frac{d^2 y_{\text{real}}}{dt^2} - \left(\frac{dy_{\text{real}}}{dt} \right)^2$$

Following the transfer of the learned neuronal parameters by the RNN, to the second segment where, we will compute $x(t+1)$, which is the next data point in the discrete temporal sequence; computation of which, may result in an additional data loss, represented as $\mathfrak{L}_{\text{data}}$, as in (9).

$$\mathfrak{L}_{\text{data}} = \text{RMSE}(\mathfrak{X}_{\text{pred}}, \mathfrak{X}_{\text{real}}) \quad (9)$$

$$\text{where, } \mathfrak{X}_{\text{pred}} = \frac{d^0(x_{\text{pred}}(t+1))}{dt^0}, \frac{d^1(x_{\text{pred}}(t+1))}{dt^1}, \frac{d^2(x_{\text{pred}}(t+1))}{dt^2}, \text{ and}$$

$$\mathfrak{X}_{\text{real}} = \frac{d^0(x_{\text{real}}(t+1))}{dt^0}, \frac{d^1(x_{\text{real}}(t+1))}{dt^1}, \frac{d^2(x_{\text{real}}(t+1))}{dt^2}.$$

Algorithm 1 presents the Pseudo code for NwPiNN, and Fig. 3 presents the pictorial representation of the architecture for NwPiNN model. In the context of the Pseudo code, in Algorithm 1, for the pre-training as well as training phase, the loss functions were combined with data losses as $\mathfrak{L} = \mathfrak{L}_{\text{data}} + C_1 \times \mathfrak{L}_{\text{phy}}$ and $\mathfrak{L} = \mathfrak{L}_{\text{data}} + C_2 \times \mathfrak{L}_{\text{phy}}$, respectively, where C_1 and C_2 are the hyperparameters. For the Newtonian physics informed neural network (NwPiNN), we considered both these hyperparameters to be 0.2. The value is chosen through Trial and Error. Now, since in this research, we are working with spatio-temporal data, we would apply the NwPiNN model recursively for all the $q \times p$ pixels of the lag frames.

Algorithm 1 Newtonian physics informed neural network (NwPiNN).

Require: $\{x(t-i)\}_{i=1}^{|x|}$, $\{\frac{d}{dt}x(t-i)\}_{i=1}^{|x|}$, and $\{\frac{d^2}{dt^2}x(t-i)\}_{i=1}^{|x|}$

Ensure: $x(t+1)$

```

1: function NwPiNN
2:   Initialize the Weights ( $W_i$ ) and Biases ( $B_i$ )
3:   if Pre-training then
4:     for  $\epsilon = 1$  to  $\epsilon_{\text{max}}$  do
5:       if Termination Condition == TRUE then
6:         break
7:       end if
8:       Get  $x(t+1)$ ,  $\frac{d}{dt}(x(t+1))$ ,  $\frac{d^2}{dt^2}(x(t+1))$  from NwPiNN
9:       Backpropagate as  $\mathfrak{L} = \mathfrak{L}_{\text{data}} + C_1 \times \mathfrak{L}_{\text{phy}}$ 
10:      Store the Weights ( $W_i$ ) and Biases ( $B_i$ )
11:    end for
12:  else
13:    for  $\epsilon = 1$  to  $\epsilon_{\text{max}}$  do
14:      if Termination Condition == TRUE then
15:        break
16:      end if
17:      Load the Weights ( $W_i$ ) and Biases ( $B_i$ )
18:      Get  $x(t+1)$ ,  $\frac{d}{dt}(x(t+1))$ ,  $\frac{d^2}{dt^2}(x(t+1))$  from NwPiNN
19:      Backpropagate as  $\mathfrak{L} = \mathfrak{L}_{\text{data}} + C_2 \times \mathfrak{L}_{\text{phy}}$ 
20:    end for
21:  end if
22: end function

```

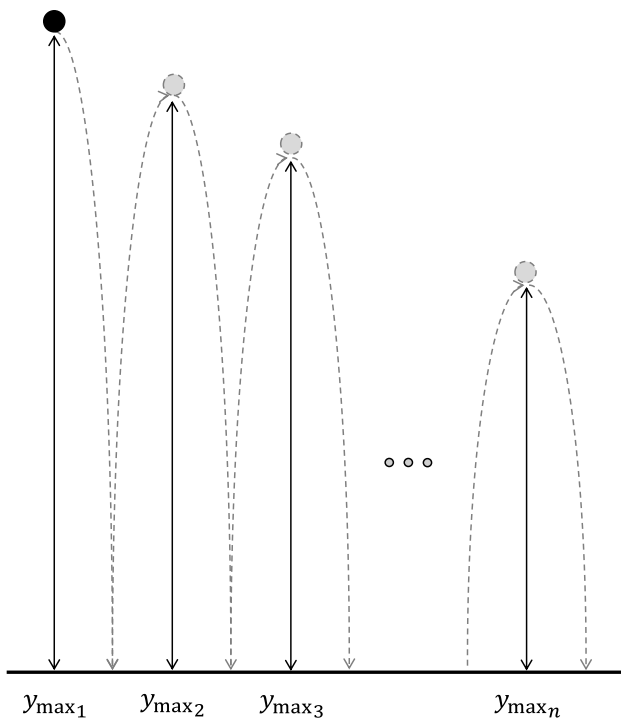


Fig. 4 Bouncy ball synthetic dataset

5 Experimental Setup

This section discusses the data and metrics for the experiment and finally delineates the results.

5.1 Datasets

For testing the efficiency of our model, we would run the model of 2 synthetic datasets,

1. *Repeated projectile* on an elastic surface, which is a 200 frame code generated synthetic dataset, generated subjecting to 3 degrees of freedom, initial velocity (v_0), angle of projection (Θ), and coefficient of restitution (e_{res}). Varying any (or all) of these parameters is expected to have a visible influence on the Number of Repetition (n_{rep}), and the Maximum Height (y_{max_i}) for these n_{rep} repeated projectiles, $\forall 1 \leq i \leq n_{rep}$.

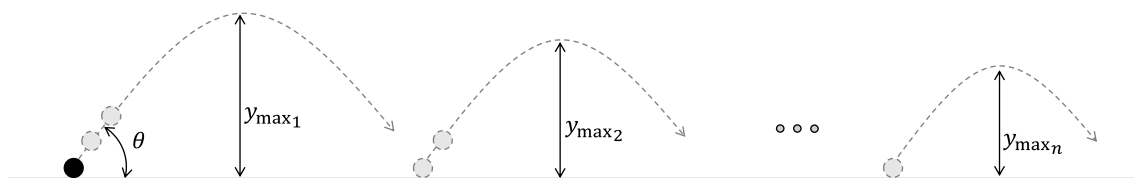


Fig. 5 Projectile synthetic dataset

2. *Bouncy ball* on a hard pitch, is again a 200 frame code generated synthetic dataset, generated subjecting to 2 degrees of freedom, initial height (y_{max_1}), and coefficient of restitution (e_{res}). Varying one (or both) of these parameters is expected to have a visible influence on the number of bounce (n_{bounce}), and the maximum height (y_{max_i}) for n_{bounce} repeated projectiles, $\forall 2 \leq i \leq n_{bounce}$.

Figures 5, and 4 give a pictorial representation of the Synthetic Datasets.

5.2 Implementation

Newtonian physics informed neural network (NwPiNN) is implemented on a general MIPS-5 Stage Architecture, with an Intel Core i7-1065G7 microprocessor, an NVIDIA GeForce MX330 graphic accelerator, and an Intel(R) Iris(R) Plus Graphics graphic accelerator.

5.3 Metrics

For evaluation of Newtonian physics informed neural network (NwPiNN), we would use the following metrics,

1. *Root mean squared error*, which calculates the squared root of the average squared difference between the estimated values ($\gamma_{i,j,n+1}^{pred}$) and the actual gray values ($\gamma_{i,j,n+1}^{real}$) for the $p \times q$ pixels. Mathematically,

$$RMSE = \sqrt{\frac{\sum_{i=1}^p \sum_{j=1}^q (\gamma_{i,j,n+1}^{pred} - \gamma_{i,j,n+1}^{real})^2}{p \times q}}$$

2. *Mean absolute error*, which calculates the average absolute difference between the estimated values ($\gamma_{i,j,n+1}^{pred}$) and the actual gray values ($\gamma_{i,j,n+1}^{real}$) for the $p \times q$ pixels. Mathematically,

$$MAE = \frac{\sum_{i=1}^p \sum_{j=1}^q |\gamma_{i,j,n+1}^{pred} - \gamma_{i,j,n+1}^{real}|}{p \times q}$$

5.4 Results

The NwPiNN Module is implemented on the synthetic datasets, as discussed in Subsection 5.1. Table 1 gives a tabulated view of the RMSE, and MAE of the predicted visual data, and the actual visual data. The results were compared with existing models, like ConvLSTM [18], and CNN-LSTM [22], and are tabulated in Table 2. For each tabulated data, we have done an extensive run of 5 counts for each arrangement, and their mean values are reported, with the standard deviation from mean in brackets.

Table 1 Tabularized results for the implementation of NwPiNN module on the repeated projectile, and bouncy ball dataset with variable initial parameters

Dataset	Number of data points (x)	Parameters	RMSE	MAE
Repeated projectile	200	$v_0 = 5, \Theta = 45^\circ, e_{res} = 0.5$	4.126 (0.259)	3.148 (0.128)
		$v_0 = 6, \Theta = 45^\circ, e_{res} = 0.6$	5.249 (0.378)	3.551 (0.201)
		$v_0 = 7, \Theta = 45^\circ, e_{res} = 0.7$	4.825 (0.895)	4.002 (0.168)
Bouncy ball	200	$y_{max} = 5, e_{res} = 0.5$	2.589 (0.137)	2.222 (0.271)
		$y_{max} = 6, e_{res} = 0.6$	3.042 (0.199)	3.010 (0.241)
		$y_{max} = 7, e_{res} = 0.7$	3.236 (0.583)	3.164 (0.946)

The module is run recursively on the same datasets up to a count of 5, and their respective RMSE, MAE and their deviation from the mean have been noted

Table 2 Comparative analysis of the proposed NwPiNN model with the pre-existing CONV-LSTM, and CNN-LSTM models

Datasets with parameters	NwPiNN		CONV-LSTM		CNN-LSTM	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Repeated projectile, $v_0 = 5, \Theta = 45^\circ, e_{res} = 0.5$	4.126 (0.259)	3.148 (0.128)	5.077 (0.175)	4.519 (0.206)	5.109 (0.194)	3.997 (0.212)
Repeated projectile, $v_0 = 6, \Theta = 45^\circ, e_{res} = 0.6$	5.249 (0.378)	3.551 (0.201)	5.937 (0.260)	5.100 (0.210)	5.549 (0.276)	4.528 (0.277)
Repeated projectile, $v_0 = 7, \Theta = 45^\circ, e_{res} = 0.7$	4.825 (0.895)	4.002 (0.168)	4.597 (0.349)	3.803 (0.283)	4.978 (0.334)	4.185 (0.512)
Bouncy ball, $y_{max} = 5, e_{res} = 0.5$	2.589 (0.137)	2.222 (0.271)	2.993 (0.203)	2.952 (0.326)	3.948 (0.265)	3.592 (0.200)
Bouncy ball, $y_{max} = 6, e_{res} = 0.6$	3.042 (0.199)	3.010 (0.241)	3.146 (0.129)	3.184 (0.194)	4.251 (0.206)	4.072 (0.337)
Bouncy ball, $y_{max} = 7, e_{res} = 0.7$	3.236 (0.583)	3.164 (0.946)	3.347 (0.159)	3.238 (0.585)	3.937 (0.448)	4.003 (0.573)

The best results for each dataset and initial parameters have been marked in bold

6 Discussion

This section is an introspection on the merits and demerits of the proposition, NwPiNN from a readers’ point of view. While, the need of the hour is machine intelligence, data engineering is quite an important task, because, as per expert opinions the 4th industrial revolution will come into the hands of artificial intelligence, and data will play a pivotal role in that. Concerning the proposal laid in this research, the PiNN, induced by the laws of Newtonian physics has proved to work efficiently for the prediction of image sequences (in layman’s terms). On a broader note, though the task performed is time-series forecasting, making use of standalone models like LSTM, or transformers won’t be able to take up the variation in every derivate of pixel displacement; these

models can forecast the zeroth order differentials of the base pixel displacement, it won't be able to catch up with the velocity, acceleration, and higher differentials, and for that, we would need some physics informed neural network that can model them based on some physical laws. For the same, NwPiNN comes into action by consuming the modelling capabilities of LSTM and the laws of classical physics. But, since everything good comes at a cost, for this model the major challenge (or limitation) is its dependence on computational units (or the semiconductor industry). As a testing phase, we implemented the model on a general MIPS-5 Stage Architecture, with an Intel Core i7-1065G7 microprocessor, an NVIDIA GeForce MX330 graphic accelerator, and an Intel(R) Iris(R) Plus Graphics graphic accelerator. Still, on a large scale, when dealing with some very large-scale data sets, of higher dimensionalities, computational requirements will skyrocket. Besides, for testing purposes, a synthetic dataset is used, thus the model performance may fluctuate practically, though a promising expected performance, similar to the results described in Sect. 5.4 can be anticipated.

7 Conclusion

Nothing in this entire universe is complete, and so is the scenario with our proposition, Newtonian physics informed neural network (NwPiNN). For a wide variety of cases, the proposed model performs well in comparison with the pre-existing models for time series forecast of visual data, for instance, ConvLSTM, CNN-LSTM, etc. to name a few. Due to computational constraints (or limitations), we considered visual data of 200×200 pixels, but NwPiNN can be applied to even a broader spectrum of visuals, providing good computational power with GPU units.

A future direction to this research could be on optimizing the algorithm to a computationally less expensive model, or ensembling the same with lightweight models like NBEATS, NHITS, or transformers. In a nutshell, the model has its arms spread throughout the ever-evolving domain of Sci-ML (*scientific machine learning*). To promote more study and replication to develop time series prediction in complex systems, we made the source codes and datasets available at <https://github.com/Anurag-Dutta/nwpiinn>.

Acknowledgements Not applicable.

Author Contributions Section-wise enumerated Author's Contribution is added below: Anurag Dutta: Introduction (Sect. 1), Problem Statement (Sect. 2), Newtonian Physics Informed Neural Network (NwPiNN) (Sect. 4). K. Lakshmanan: Preliminaries (Sect. 3), and Conclusion (Sect. 7). Sanjeev Kumar: Experimental Setup (Sect. 5), and Discussion (Sect. 6). A. Ramamoorthy: Conclusion (Sect. 7)

Funding The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

Availability of Data and Materials The source codes and datasets are available at <https://github.com/Anurag-Dutta/nwpiinn>.

Declarations

Conflict of Interest The above manuscript is subjected to no conflict of interest or any conflict of interest.

Ethical Approval Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Arslan S, Schmidt J, Bass C, et al. A systematic pan-cancer study on deep learning-based prediction of multi-omic biomarkers from routine pathology images. *Commun Med*. 2024;4(1):48.
2. Baum EB. On the capabilities of multilayer perceptrons. *J Complex*. 1988;4(3):193–215.
3. Bebis G, Georgiopoulos M. Feed-forward neural networks. *IEEE Potentials*. 1994;13(4):27–31.
4. Cai S, Mao Z, Wang Z, et al. Physics-informed neural networks (pinns) for fluid mechanics: a review. *Acta Mech Sin*. 2021;37(12):1727–38.
5. Cerqueira V, Torgo L, Bontempi G. Instance-based meta-learning for conditionally dependent univariate multi-step forecasting. *Int J Forecast*. 2024.
6. Chakraborty T, Reddy KS U, Naik SM, et al. Ten years of generative adversarial nets (gans): a survey of the state-of-the-art. *Mach Learn Sci Technol*. 2023.
7. Challu C, Olivares KG, Oreshkin BN, et al. Nhits: neural hierarchical interpolation for time series forecasting. In: *Proceedings of the AAAI conference on artificial intelligence*. 2023. p. 6989–97.
8. Chang Y, Wang X, Wang J, et al. A survey on evaluation of large language models. *ACM Trans Intell Syst Technol*. 2023.
9. Davis MH. *Markov models & optimization*. Milton Park: Routledge; 2018.
10. De Gooijer JG, Hyndman RJ. 25 years of time series forecasting. *Int J Forecast*. 2006;22(3):443–73.
11. Dongare A, Kharde R, Kachare AD, et al. Introduction to artificial neural network. *Int J Eng Innov Technol (IJEIT)*. 2012;2(1):189–94.
12. Dutta A, Panja M, Kumar U, et al. Van der pol-informed neural networks for multi-step-ahead forecasting of extreme climatic events. In: *NeurIPS 2023 AI for science workshop*. 2023.

13. Elabid Z, Chakraborty T, Hadid A. Knowledge-based deep learning for modeling chaotic systems. In: 21st IEEE international conference on machine learning and applications (ICMLA). IEEE; 2022. p. 1203–9.
14. Gardner ES Jr. Exponential smoothing: the state of the art. *J Forecast.* 1985;4(1):1–28.
15. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 1997;9(8):1735–80.
16. Just J. Natural language processing for innovation search-reviewing an emerging non-human innovation intermediary. *Technovation.* 2024;129:102,883.
17. Li ZN, Drew MS, Liu J. *Fundamentals of multimedia.* Cham: Springer; 2004.
18. Lin Z, Li M, Zheng Z, et al. Self-attention convlstm for spatio-temporal prediction. In: *Proceedings of the AAAI conference on artificial intelligence.* 2020. p. 11,531–8.
19. Liu C, Jin Y, Xu K, et al. Beyond short-term snippet: video relation detection with spatio-temporal global context. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2020. p. 10,840–9.
20. Medsker L, Jain LC. *Recurrent neural networks: design and applications.* Boca Raton: CRC Press; 1999.
21. Meyer M, Langer A, Mehlretter M, et al. Image-based deep learning for the time-dependent prediction of fresh concrete properties. 2024. [arXiv:2402.06611](https://arxiv.org/abs/2402.06611).
22. Mutegeki R, Han DSA. cnn-lstm approach to human activity recognition. In: *2020 international conference on artificial intelligence in information and communication (ICAIC).* IEEE; 2020. p. 362–6.
23. Nelson BK. Time series analysis using autoregressive integrated moving average (arima) models. *Acad Emerg Med.* 1998;5(7):739–44.
24. Newton I. *Philosophiae naturalis principia mathematica, vol. 1.* G. Brookman; 1833.
25. Ning X, Jia L, Wei Y, et al. Epi-dnns: epidemiological priors informed deep neural networks for modeling covid-19 dynamics. *Comput Biol Med.* 2023;158:106,693.
26. Olivares KG, Challu C, Marcjasz G, et al. Neural basis expansion analysis with exogenous variables: forecasting electricity prices with nbeatsx. *Int J Forecast.* 2023;39(2):884–900.
27. Oreshkin BN, Carpov D, Chapados N, et al. N-beats: neural basis expansion analysis for interpretable time series forecasting. 2019. [arXiv:1905.10437](https://arxiv.org/abs/1905.10437).
28. Panja M, Chakraborty T, Nadim SS, et al. An ensemble neural network approach to forecast dengue outbreak based on climatic condition. *Chaos Solitons Fractals.* 2023;167:113,124.
29. Reich Y. Machine learning techniques for civil engineering problems. *Comput Aided Civ Infrastruct Eng.* 1997;12(4):295–310.
30. Rodríguez A, Cui J, Ramakrishnan N, et al. Einns: epidemiologically-informed neural networks. In: *Proceedings of the AAAI conference on artificial intelligence;* 2023. p. 14,453–60.
31. Rusmawati Y. Automated reasoning on machine learning model of legislative election prediction. In: *Proceedings of the 10th international joint conference on knowledge graphs;* 2021. p. 200–4.
32. Tanaka G, Yamane T, Héroux JB, et al. Recent advances in physical reservoir computing: a review. *Neural Netw.* 2019;115:100–23.
33. Torrey L, Shavlik J. Transfer learning. In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques.* IGI global; 2010. p. 242–64.
34. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. *Adv Neural Inf Process Syst.* 2017. p. 30.
35. Vladimirov VS, Volovich IV. *Differential calculus.* *Teor Mat Fiz.* 1984;59(1):3–27.
36. Zhou Y, Du N, Huang Y, et al. Brainformers: trading simplicity for efficiency. In: *International conference on machine learning.* PMLR; 2023. p. 42,531–42.

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