



A Comparative study of Neural Networks vs. Mechanistic ODE Models for Health Data Forecasting

Deepayan Das, Cécile Proust-Lima, Mélanie Prague, Xavier Hinaut

► To cite this version:

Deepayan Das, Cécile Proust-Lima, Mélanie Prague, Xavier Hinaut. A Comparative study of Neural Networks vs. Mechanistic ODE Models for Health Data Forecasting. Centre Inria de l'Université de Bordeaux; Inserm. 2023. hal-04935796

HAL Id: hal-04935796

<https://inria.hal.science/hal-04935796v1>

Submitted on 7 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

A Comparative study of Neural Networks vs. Mechanistic ODE Models for Health Data Forecasting

Deepayan Das, MNEMOSYNE team, Bordeaux University, Inria Center of Bordeaux University, LaBRI, IMN
Cécile Proust-Lima*, BIOSTAT team, Bordeaux Population Health Research Center, Inserm
Mélanie Prague*, SISTM team, Inria Center of Bordeaux University
Xavier Hinault*, MNEMOSYNE team, Inria Center of Bordeaux University, LaBRI, IMN

CONTENTS

I	Introduction	1
II	Problem Definition and Underlying Methodology	1
III	Architectures Used	3
III-A	Multi Layer Perceptron	3
III-B	LSTM	3
III-C	Echo State Networks	3
IV	Evaluation Metrics	3
IV-A	Bias	4
IV-B	Relative Bias	4
IV-C	Mean Squared Error (MSE)	4
IV-D	Relative Mean Squared Error (rMSE)	4
V	Results and Discussion	4
V-A	Bias	4
V-B	Relative Bias	4
V-C	Mean Squared Error (RMSE)	4
V-D	Overall Observations	4
VI	Qualitative Results and discussions	5
VII	Conclusion and Future Works	6
References		6

LIST OF FIGURES

1	The above figure demonstrates the simulated training data for patients belonging to group 0 and group 1. The scatter plot shows the data with missing values and added noise. As we can observe that after removing data at random, we are left with very sparse and noisy data. Thus to make the data continuous, we perform data imputation for both groups shown by red solid and dashed lines. It is worthy to note that that post-imputation and smoothing, the data retains a noisy nature as opposed to more smooth trajectories predicted by the mechanistic ODE model shown in solid and dashed orange lines.	3
2	Above figure demonstrates the qualitative results from our sequential model. The top row depicts the true vs predicted trajectories from our ESN model while the bottom row true vs predicted trajectories from our LSTM model.	5
3	True trajectory vs predicted trajectory from a Mechanistic ODE model	6

LIST OF TABLES

I	Bias results for different models and inputs.	4
II	Relative Bias results for different models and inputs.	4
III	RMSE of different models on different days.	4

*Corresponding authors who co-supervised the study

A Comparative study of Neural Networks vs. Mechanistic ODE Models for Health Data Forecasting

Abstract—Neural networks are known as universal approximators i.e. with sufficient data they can model any function. They have achieved exceptional results in a variety of domains including image classification, speech-processing, and natural language generation, serving as state-of-the-art models for classification and regression tasks. However, one of the main limitations of such models are that they require a large amount of data to accomplish any given task. The availability of data becomes even more challenging in the domain of health care. In our study we highlight this limitation by comparing the performance of 3 distinct neural network architectures: an MLP, an ESN model with a reservoir, and an LSTM model against a mechanistic ODE model on a simple data forecasting task, using simulated patient data enriched with random missing values to mimic real-world scenarios.

Given the sparse nature of the data, we performed various pre-processing steps such as imputation using SAITS and smoothing with a moving average to fill in the gaps. We also used group information to further enhance the data. Reservoir and LSTM models were trained auto-regressively, while the MLP model leveraged initial data to make predictions at specific future points.

The evaluation metrics—bias, relative bias, MSE, and relative RMSE—revealed the superior performance of the Mechanistic model in handling sparse data, registering a minimal bias of 0.003 and an RMSE of 0.09 on the 180th day.

In conclusion, the study illustrates that Mechanistic ODE models hold a comparative advantage in forecasting under conditions of limited and sparse data, offering insights into potential enhancements for Neural Network models in data-restricted environments.

I. INTRODUCTION

Neural Networks have revolutionized the field of machine learning, achieving remarkable success across a wide range of tasks including image classification [4], natural language processing [6], and speech recognition [1]. Given sufficient data Neural Networks can model a vast array of complex functions. However, these models are not without their limitations, particularly in domains where data is scarce or expensive to acquire, such as in healthcare and medical analytics.

In the healthcare domain, the collection and annotation of data are notoriously challenging and resource-intensive, often rendering the data-driven approach of Neural Networks sub-optimal. This limitation is pronounced in tasks like forecasting health-related data, where models are expected to make reliable predictions based on limited and sometimes noisy input data. In our endeavor to illustrate this challenge, we focus on a task related to HIV forecasting. In this task, models are provided

with seed data and are expected to predict the viral load of patients at the end of six months, a process rife with inherent complexities due to the variability in disease progression and response to treatment among individuals.

For this comparative study, we delve into the performance of three diverse Neural Network architectures: a) a Multilayer Perceptron (MLP), renowned for its simplicity and effectiveness in capturing linear relationships, b) a Long Short-Term Memory network (LSTM) [2], distinguished for its ability to model sequential data by mitigating the vanishing gradient problem, and c) an Echo State Network (ESN) [3] model, characterized by its reservoir computing capability which allows it to learn complex temporal patterns. These models are compared against a Mechanistic ODE model, renowned for its efficacy in modeling dynamic systems, and a Spline Regression model, which is instrumental in capturing non-linear relationships in the data through piecewise polynomials.

To critically assess the efficacy of these models, we systematically compared their predictions at regular intervals, every 30 days, until the culmination of the six-month period. Our empirical examination revealed a consistent superior performance of the Mechanistic ODE model over the Neural Network models in forecasting the viral load, emphasizing the importance of incorporating domain knowledge and dynamic system modeling, especially when the available data is sparse and constrained.

This study, while highlighting the constraints and challenges associated with deploying Neural Networks in data-limited scenarios like healthcare, also sheds light on the comparative advantages of different modeling approaches. It underscores the necessity of exploring alternative models and methodologies, such as Mechanistic ODE models and Spline Regression, which can leverage domain knowledge and inherent system dynamics to make reliable predictions in the face of data limitations.

Through this comprehensive exploration, we aim to contribute valuable insights to the ongoing discourse in the realm of healthcare analytics and to prompt further investigation into the development of more robust and data-efficient models capable of navigating the intricate landscape of medical data forecasting.

II. PROBLEM DEFINITION AND UNDERLYING METHODOLOGY

Let's denote our simulated training dataset as D_{train} . It consists of data from 100 patients, with 50 patients belonging to

Group 0 and the remaining 50 belonging to Group 1, spanning a period of 6 months. The data collection is performed weekly, with a variance of ± 1 or ± 2 days, resulting in approximately 22 to 24 readings per patient, expressed as:

$$D_{train} = (x_{i1}, g_1), (x_{i2}, g_2), \dots, (x_{i100}, g_{100})$$

Here x_{ik} and g_k represents the irregularly taken readings and g_k represents the group information for the k^{th} patient.

For validation and testing, we simulate a separate dataset, denoted as D_{test} , again consisting of patients categorized into Group 0 and Group 1. This dataset, akin to the training setup, includes readings obtained within the initial 3 weeks, approximately 3 to 4 readings per patient, with inherent missing values attributed to instances where patients missed their scheduled check-ups. Formally, the test dataset can be represented as: We follow a same nomenclature for the test data D_{test} :

$$D_{test} = (x_{j1}, g_1), (x_{j2}, g_2), \dots, (x_{j100}, g_{100})$$

However, as mentioned before for D_{train} we consider readings over a period of 180 days while for D_{test} we consider a time-period of 21 days. Also, it is essential to note that x_{ik} and x_{jk} are sequences of data captured at irregular time interval for patient k and not a single data point.

We perform imputation on both D_{train} and D_{test} to make the sequence continuous which is expressed as:

$$D_{train, imp} = \{(f_{imp}(x_{i1}), g_1), (f_{imp}(x_{i2}), g_2), \dots, (f_{imp}(x_{i100}), g_{100})\}$$

$$D_{test, imp} = \{(f_{imp}(x_{j1}), g_1), (f_{imp}(x_{j2}), g_2), \dots, (f_{imp}(x_{j100}), g_{100})\}$$

To elucidate the training procedure for sequence-based models, particularly for Reservoir and RNN models, we consider the imputed dataset $D_{train, imp}$ and $D_{test, imp}$ representing all patients. However, for the sake of clarity in illustrating the auto-regressive training method, we will narrow our focus to the dataset corresponding to a single patient. Thus, let's denote the imputed data sequence for the k^{th} patient as $(x_{k,1:T}, g_k)$, where T is the total number of time points in the sequence.

The training of sequence-based models adopts an auto-regressive approach, where the model, at each time step t , is exposed to the sequence of observed values up to that time point and is trained to predict the value at the next time point, $t + 1$. Formally, the auto-regressive training for a sequence-based model can be illustrated as follows:

$$\hat{x}_{k,t+1} = f_{seq}(x_{k,1:t}, g_k; \theta) \quad (1)$$

Here:

- $\hat{x}_{k,t+1}$ represents the model's prediction for the k^{th} patient at time $t + 1$.
- f_{seq} symbolizes the sequence-based model being utilized, e.g., an RNN or a Reservoir model.
- $x_{k,1:t}$ is the sequence of observed values for the k^{th} patient up to time t .

- g_k denotes the group information for the k^{th} patient.
- θ are the model parameters being optimized during training.

The objective during the training phase is to optimize the model parameters, θ , to minimize the difference between the predicted values, $\hat{x}_{k,t+1}$, and the actual observed values, $x_{k,t+1}$, across all time points for the selected patient in the imputed training dataset, $D_{train, imp}$.

During the inference phase, the process is initiated by providing the models with the seed data, denoted as $D_{test, imp}$. The models are then tasked with generating sequences representing the forthcoming 180 days. In this autoregressive generation process, the output from the models at each time step is subsequently fed as input for generating data for the next time step. Formally, for any sequence-based model f_{seq} with parameters θ_{seq} , the generation process can be represented as:

$$\hat{x}_{k,t+1} = f_{seq}(\hat{x}_{k,1:t}, g_k; \theta_{seq}), \quad t = 21, 22, \dots, 179 \quad (2)$$

This approach ensures a continuous, iterative generation of the sequence, simulating the potential progression over the span of 180 days using the provided seed data from $D_{test, imp}$.

Unlike the sequence-based models, the MLP, a non-sequence-based model, is trained in a different manner to illustrate the distinctive advantages of sequence-based models for tasks of this nature. Given the constraints during the inference phase, where only three weeks of data are accessible, we train the MLP with three weeks of data and leverage it to forecast the viral loads on predefined days, specifically at the end of every month.

Formally, we train the MLP model, f_{mlp} , using the initial three weeks of imputed training data, $x_{k,1:21}$ and group information g_k for the k^{th} patient, to predict the viral loads, $\hat{x}_{k,\tau}$, on the predefined days, τ , representing the end of every month:

$$\hat{x}_{k,\tau} = f_{mlp}(x_{k,1:21}, g_k; \theta_{mlp}) \quad (3)$$

Here:

- $\hat{x}_{k,\tau}$ denotes the predicted viral loads for the k^{th} patient on day τ .
- f_{mlp} represents the MLP model.
- $x_{k,1:21}$ is the sequence of observed values for the k^{th} patient during the initial three weeks.
- g_k symbolizes the group information for the k^{th} patient.
- θ_{mlp} are the model parameters being optimized during training.

The underlying objective is to optimize the MLP model parameters, θ_{mlp} , to minimize the discrepancy between the predicted viral loads, $\hat{x}_{k,\tau}$, and the actual observed values, $x_{k,\tau}$, on the predefined days in the imputed training dataset, $D_{train, imp}$.

For both sequence and non-sequence based models we use mean-square error as our loss function to optimize the model parameters.

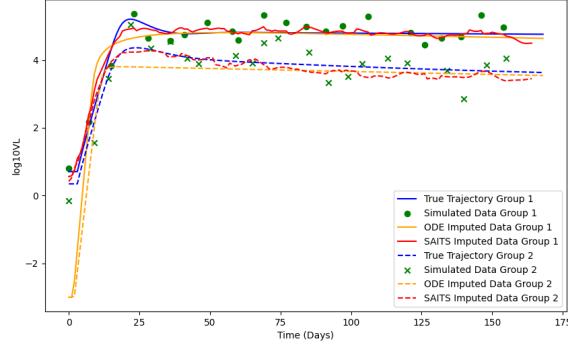


Fig. 1: The above figure demonstrates the simulated training data for patients belonging to group 0 and group 1. The scatter plot shows the data with missing values and added noise. As we can observe that after removing data at random, we are left with very sparse and noisy data. Thus to make the data continuous, we perform data imputation for both groups shown by red solid and dashed lines. It is worthy to note that that post-imputation and smoothing, the data retains a noisy nature as opposed to more smooth trajectories predicted by the mechanistic ODE model shown in solid and dashed orange lines.

III. ARCHITECTURES USED

As mentioned in our previous sections we made use of both sequence and non-sequence based models. Let's delve into each.

A. Multi Layer Perceptron

Multilayer Perceptrons (MLPs) are the most straightforward models utilized in our study. We categorize them as non-sequential models due to their inability to account for the sequential nature of the input data during the prediction phase. The architecture of the implemented MLP model is composed of two linear layers, each containing 512 neurons. These layers are interspersed with a ReLU non-linearity, represented as:

$$f_{\text{MLP}}(x) = \text{ReLU}(W_1x + b_1)W_2 + b_2 \quad (4)$$

Here, W_1 , W_2 , b_1 , and b_2 are the weights and biases of the layers, and x represents the input data. The choice of ReLU as the activation function is motivated by its proven efficiency and the sparsity it induces.

The output layer of our model is tailored to have a dimension of 6. This design choice is based on our objective of evaluating the model's predictions at the conclusion of every month over a half-year period, allowing for a comprehensive insight into the model's forecasting capabilities across distinct timeframes.

B. LSTM

We employed a unidirectional Long Short-Term Memory (LSTM) network, complemented with a Multilayer Perceptron

(MLP) as its output layer. LSTMs are an advanced variant of the standard Recurrent Neural Networks (RNNs). The distinctive feature of RNNs, as opposed to simpler models like MLPs, is their ability to incorporate outputs from previous time steps in conjunction with the current input. This intrinsic property enables them to model sequential data effectively, making them especially proficient in tasks where the temporal dimension is crucial.

The mathematical formulation of a basic RNN can be represented as:

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \quad (5)$$

$$y_t = W_{hy}h_t + b_y \quad (6)$$

Here, h_t is the hidden state at time t , and y_t is the output at time t . The W and b parameters represent the weights and biases, respectively, and σ denotes the activation function, typically a hyperbolic tangent or a sigmoid function. x_t represents the input at time t , and the subscripted hh , xh , and hy signify the associations between the hidden-hidden, input-hidden, and hidden-output layers, respectively.

C. Echo State Networks

For our third model, we utilized an Echo State Network (ESN) complemented by a Ridge Regressor for the output layer. The reservoir, a dynamic, recurrently connected hidden layer, is composed of 512 neurons. A unique feature of ESNs is the adjustment of only the output layer's weights during the training, leaving the reservoir's weights unaltered. This results in a reduction in computational overhead and expedites the training process. The Ridge Regressor mitigates overfitting by imposing a penalty on the magnitude of the coefficients.

The reservoir dynamics can be represented as:

$$\mathbf{r}(t) = \tanh(\mathbf{W}_{\text{in}}\mathbf{u}(t) + \mathbf{Wr}(t-1)) \quad (7)$$

Where $\mathbf{r}(t)$ denotes the state of the reservoir neurons at time t , \mathbf{W}_{in} represents the input weight matrix, $\mathbf{u}(t)$ is the input at time t , and \mathbf{W} is the reservoir weight matrix.

The network output is computed as:

$$\mathbf{y}(t) = \mathbf{W}_{\text{out}}\mathbf{r}(t) \quad (8)$$

Here, $\mathbf{y}(t)$ is the output at time t and \mathbf{W}_{out} is the output weight matrix, which is adjusted using Ridge Regression:

$$\mathbf{W}_{\text{out}} = (\mathbf{R}^{\top}\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{R}^{\top}\mathbf{D} \quad (9)$$

In this equation, \mathbf{R} is the reservoir state matrix, \mathbf{D} is the desired output matrix, λ is the regularization coefficient, and \mathbf{I} is the identity matrix.

IV. EVALUATION METRICS

To quantitatively assess the performance of our models, we employed a set of metrics designed to measure different aspects of forecasting accuracy. These metrics include bias, relative bias, Mean Squared Error (MSE), and relative Root Mean Squared Error (RMSE). Below, we provide formal definitions of these metrics.

A. Bias

Bias is a measure of the systematic error introduced by the modeling process. It is calculated as the average difference between the predicted values and the true values:

$$\text{Bias} = \frac{1}{N} \sum_{i=1}^N (P_i - O_i) \quad (10)$$

Where N is the number of observations, P_i are the predicted values, and O_i are the observed true values.

B. Relative Bias

Relative bias provides a normalized measure of the systematic error, expressed as a percentage of the true value:

$$\text{Relative Bias} = \frac{1}{N} \sum_{i=1}^N \left(\frac{P_i - O_i}{O_i} \right) \times 100 \quad (11)$$

C. Mean Squared Error (MSE)

MSE is a widely-used metric for regression tasks, measuring the average of the squares of the errors between predicted and true values:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (P_i - O_i)^2 \quad (12)$$

D. Relative Mean Squared Error (rMSE)

Relative Mean Squared Error normalizes the MSE value by the mean of the observed true values, providing a relative measure of error:

$$\text{rMSE} = \frac{\frac{1}{N} \sum_{i=1}^N (P_i - O_i)^2}{\frac{1}{N} \sum_{i=1}^N O_i^2} \times 100 \quad (13)$$

These metrics enabled us to gain insights into both the accuracy and the reliability of our models in forecasting health-related data.

V. RESULTS AND DISCUSSION

In this section, we discuss the outcomes of the different models and inputs based on the metrics of Bias, Relative Bias, and Relative Mean Squared Error (RMSE). The results are summarized in Tables I, II, and III.

Method	Input	30	60	90	120	150	180
Reservoir	Raw	0.3699	0.1610	0.1013	0.0522	0.0113	-0.0102
	ODE	0.2940	0.0693	0.0090	-0.0406	-0.0820	-0.1038
LSTM	Raw	0.2881	0.1891	0.1328	0.0834	0.0419	0.0198
	ODE	0.4376	0.1961	0.1361	0.0866	0.0451	0.0230
MLP	Raw	0.1574	0.0431	0.1247	0.0089	-0.0669	0.2348
	ODE	0.1806	0.0078	-0.0033	-0.0111	-0.0121	-0.0232
Mechanistic	Raw	0.2919	0.0475	-0.0232	-0.0239	-0.0389	-0.0302
Spline	Raw	0.7432	-0.4665	-0.2404	0.2080	-0.0250	-0.5169

TABLE I: Bias results for different models and inputs.

Method	Input	30	60	90	120	150	180
Reservoir	Raw	0.0659	0.0362	0.0229	0.0113	0.0014	-0.0039
	ODE	0.0656	0.0164	0.0022	-0.0099	-0.0202	-0.0257
LSTM	Raw	0.0738	0.0369	0.0235	0.0118	0.0017	-0.0037
	ODE	0.098	0.0424	0.0286	0.0169	0.0068	0.0014
MLP	Raw	0.0513	0.0122	0.0315	0.0046	-0.0149	0.0302
	ODE	0.0692	0.0147	0.0055	0.0003	-0.0022	-0.0056
Mechanistic	Raw	0.0651	0.0112	-0.0056	-0.0058	-0.0096	-0.0075
	Spline	0.1658	-0.1106	-0.0578	0.0506	-0.0061	-0.1277

TABLE II: Relative Bias results for different models and inputs.

Method	Input	30	60	90	120	150	180
Reservoir	Raw	0.3598	0.2654	0.206	0.1766	0.1683	0.1703
	ODE	0.3414	0.1697	0.1531	0.1708	0.2022	0.2228
LSTM	Raw	0.2240	0.2719	0.2604	0.2254	0.2001	0.1911
	ODE	0.3623	0.1583	0.1483	0.1714	0.2045	0.2251
MLP	Raw	0.2795	0.2201	0.2555	0.2457	0.2556	0.3245
	ODE	0.308	0.1717	0.1602	0.1667	0.1741	0.202
Mechanistic	Raw	0.1736	0.0683	0.0832	0.0691	0.0869	0.0925
	Spline	0.5974	0.2392	0.0806	0.0692	0.0282	0.2975

TABLE III: RMSE of different models on different days.

A. Bias

From Table I, it can be observed that the Mechanistic model consistently outperformed the neural network-based models across different days, with the lowest bias observed being -0.0302 on the 180th day. The MLP, especially with ODE input, showed promising results with the bias reaching as low as -0.0111 on the 120th day.

B. Relative Bias

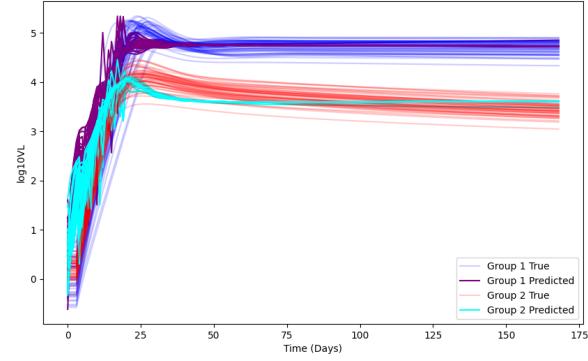
The relative bias results presented in Table II are indicative of the relative accuracy of the models. Similar to the bias results, the Mechanistic model exhibited superior performance. However, MLP with ODE input also demonstrated noteworthy precision, particularly on the 60th day, with a relative bias of -0.0033.

C. Mean Squared Error (RMSE)

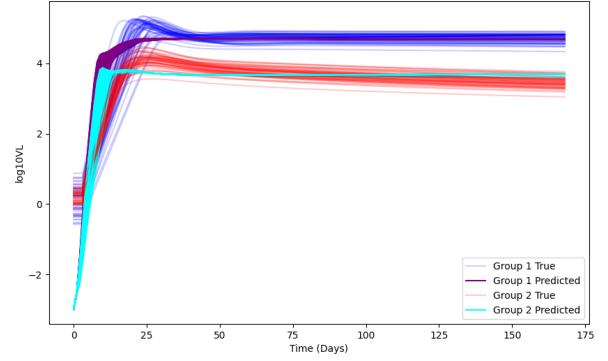
Table III outlines the RMSE results of our models. Once again, the Mechanistic model, with an RMSE of 0.09 on the 180th day, stands out, emphasizing its ability to predict with high accuracy even with sparse data. The Reservoir and LSTM models also reported competitive RMSE values, with a notable performance from the LSTM model with a RMSE of 0.165 on the 180th day with raw input.

D. Overall Observations

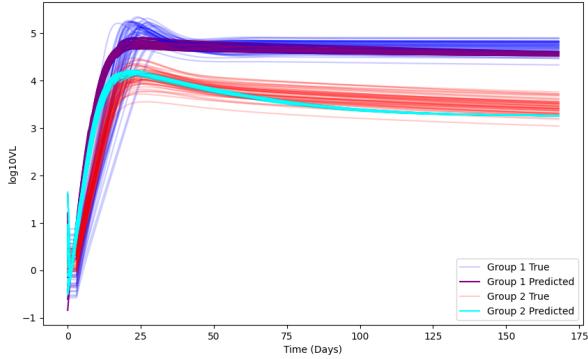
The outcomes of our study reveal a consistent trend where the Mechanistic model outshines the neural network-based models in terms of bias, relative bias, and RMSE across



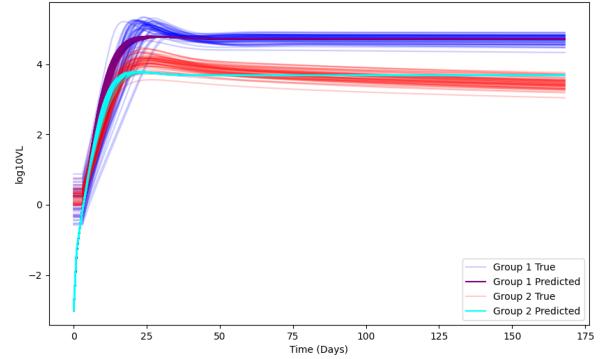
(a) True Trajectory Vs Reservoir predictions on raw data



(b) True Trajectory Vs Reservoir predictions on ODE input



(c) True Trajectory Vs LSTM predictions on raw data



(d) True Trajectory Vs Reservoir predictions on ODE data

Fig. 2: Above figure demonstrates the qualitative results from our sequential model. The top row depicts the true vs predicted trajectories from our ESN model while the bottom row true vs predicted trajectories from our LSTM model.

different days. This superior performance of the Mechanistic model is particularly highlighted in scenarios with limited and sparse data, such as in healthcare applications.

However, it is important to note that the MLP model, especially with ODE input, has shown promising results and, with further refinement and optimization, could potentially match the accuracy of the Mechanistic model in forecasting tasks.

The relative underperformance of neural network-based models in our study, specifically in sparse data scenarios, underscores the need for more advanced and tailored approaches, possibly integrating domain knowledge, to enhance the predictive accuracy of these models in healthcare applications.

VI. QUALITATIVE RESULTS AND DISCUSSIONS

In this section, we delve into a qualitative examination of the performance exhibited by our sequence-oriented models. This is achieved by comparing the actual trajectories with the outputs produced by our models.

Figure 2 presents the predictions rendered by our sequence-based model when subjected to different forms of input data, namely raw data and ODE-derived data. A side-by-side comparison is conducted, overlaying these predictions against the true trajectories. As elucidated in Section II, our approach leverages seed data to formulate the preliminary set of predictions. Beyond the seed data duration (or the so-called cut-off period), the model's output serves as the subsequent input, guiding the generation of predictions for the ensuing time step.

From our observations, it is evident that the trajectories pertaining to the two patient groups are distinguishably unique. Impressively, our sequence-based models successfully encapsulate these distinct patterns. Nonetheless, for both patient categories, the LSTM model initially tends to overestimate, diverging from the actual trajectory. This deviation diminishes as the model progresses to later time steps. Contrarily, the Echo State Network (ESN) initially offers predictions that are closely aligned with the true trajectories. However, as time progresses,

it tends to slightly overestimate. This behavior underscores the ESN model's competence in effectively modeling data, especially when data availability is sparse.

A noteworthy observation is that when confronted with imputed raw data, both the ESN and LSTM models exhibit marked differences in their predictions. However, when trained on data derived from ODEs, both models aptly grasp the ODE distribution. Their outputs, in this case, are commendably alike, highlighting their capability to adapt and replicate the underlying patterns of the ODE data.

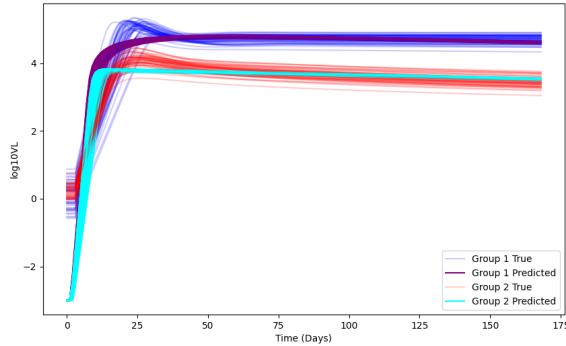


Fig. 3: True trajectory vs predicted trajectory from a Mechanistic ODE model

In Figure 3 we demonstrate the comparison of True trajectory as opposed to trajectories predicted by the Mechanistic ODE model. We can observe that the Mechanistic ODE model output are more aligned to the actual distribution compared to Reservoir or LSTM model on raw imputed data. Another observation is that when we used ODE model as an output an imputed input to train the sequence based models, both Reservoir and LSTM follow a more aligned trajectory to the true distribution. However, the disparity in predictions among individual patients is lesser in comparison to the outputs of the ODE model. We hypothesize that this reduced variation is likely why the ODE model surpassed the sequence-based models in quantitative performance.

VII. CONCLUSION AND FUTURE WORKS

In this study, we tried to assess the performance of sequence-based models in predicting patient trajectories. The quantitative and qualitative nature of our results highlighted the predictive capabilities Neural Networks and Mechanistic ODE based models in the face of sparse and limited dataset. The Mechanistic ODE model, in particular, exhibited a commendable alignment with the true trajectory, especially when compared with the Reservoir or LSTM models trained on raw imputed data. On the other hand, when sequence-based models were trained with ODE model outputs in conjunction with imputed input, their trajectories showed improved congruence with the actual distribution. However, despite these advancements, the ODE model's superior performance

in capturing individual patient variations underscores its robust quantitative capabilities.

One of the pivotal learnings from this study is the importance of capturing variation across individual patients. The ODE model's superior performance can be attributed to its adeptness in this aspect. As a future direction, our focus will be on enhancing the sequence-based models to better model the nuances and variations across individual patients. By achieving this, we aspire to bridge the performance gap between our sequence-based models and the ODE model. The promise of such advancements holds the potential to revolutionize predictive modeling in patient care, leading to more personalized and effective interventions.

ACKNOWLEDGMENTS

This study was carried out in the framework of the University of Bordeaux's France 2030 program / RRI PHDS. The library ReservoirPy [5] was used in this study: <https://github.com/reservoirpy/reservoirpy>

APPENDIX

The source code to produce these results is available on this GitHub repository: <https://github.com/Deepayan137/health-forecast>

REFERENCES

- [1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.
- [2] S Hochreiter and J Schmidhuber. Long short-term memory. *Neural Computation MIT-Press*, 1997.
- [3] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [5] Nathan Trouvain, Luca Pedrelli, Thanh Trung Dinh, and Xavier Hinaut. ReservoirPy: An efficient and user-friendly library to design echo state networks. In *Artificial Neural Networks and Machine Learning – ICANN 2020*, pages 494–505. Springer International Publishing, 2020.
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, et al. Attention is all you need [j]. In *Advances in neural information processing systems*, volume 30, pages 261–272, 2017.