A Comparative Analysis of Deep Learning and Numerical Methods for Estimating Influenza Cases

Rifat Rafiuddin (CWID: A20387294)

Mohammed Rakib (CWID: A20441817)

Submitted on:

3^{rd} *May*, 2024

Abstract

In this project, we examined and compared two primary approaches—numerical methods and machine learning models—for predicting influenza cases, specifically focusing on the H1N1 virus. Traditional numerical methods, such as the Midpoint, Modified Euler, and RK4 methods, are evaluated alongside modern machine learning techniques, including 1D Convolutional Neural Networks, Long Short-Term Memory networks, and Random Forests. The analysis demonstrates that while numerical methods provide a classic approach to modeling disease spread, machine learning models, particularly Random Forest, show superior performance in forecasting accuracy. The study utilizes real-world data from a recent H1N1 outbreak in São Paulo, Brazil, to evaluate these methods. Results indicate that machine learning models are more effective in capturing and predicting complex patterns in disease spread compared to traditional numerical models, offering valuable insights for improving public health strategies.

I. Introduction

Influenza is a highly infectious respiratory illness known for causing seasonal outbreaks and pandemics. This project explores and compares two distinct methodologies for predicting the number of influenza cases, with a specific focus on the H1N1 virus strain. Periodically, influenza poses significant public health challenges due to its rapid spread and mutation capabilities. The objective is to apply traditional numerical methods and modern machine learning and deep learning approaches to predict the virus's incidence. For the numerical approaches, the SVIR model is employed, integrating real-world data and numerical computations from a recent outbreak in São Paulo, Brazil, while the machine learning and deep learning techniques use pattern recognition to forecast trends based on historical data.

The numerical methods used in this project include the Midpoint, Modified Euler, and RK4 methods. These methods are based on classical mathematical equations to understand how influenza spreads. On the other hand, machine learning and deep learning techniques like 1D Convolutional Neural Networks (1D-CNN), Long Short-Term Memory networks (LSTM), Feature Tied LSTM (FTA-LSTM), and Random Forest rely on analyzing past data to predict future influenza outbreaks. These methods are tested for their accuracy in predicting the highest number of cases and the general accuracy of their forecasts. This helps assess how effective and understandable these methods are for modeling diseases and planning public health strategies.

II. Motivation

During class, we were introduced to various numerical approximation methods, such as the Runge-Kutta 4th Order Method, Midpoint Method, Euler's Method, and Modified Euler's Method. This sparked our interest in exploring their applications. We were particularly curious to compare these traditional methods with modern machine learning models in predicting influenza outcomes. To investigate this, we utilized a datase of weekly H1N1 influenza cases from São Paulo, Brazil provided by our instructor. We applied both classical numerical methods through the SVIR model and machine learning techniques like LSTM and Random Forests. This project aims to assess the accuracy and reliability of these methods in forecasting influenza cases, providing insights into how different computational approaches behave in approximating influenza cases.

III. Methodology

I. Dataset

The dataset employed in this study consists of weekly case records of the H1N1 influenza outbreak that occurred in São Paulo, Brazil, between 2009 and 2010. This dataset begins with an initial condition of having one infected individual at the onset of the outbreak in 2009, providing a realistic starting point for the simulation models. Over the course of the outbreak, detailed records of new influenza cases were compiled on a weekly basis, offering a granular view of the disease's spread over time.

II. SVIR Model with Numerical Methods

The SVIR model is a mathematical framework used to understand the spread of infectious diseases within a population over time. It categorizes the population into four compartments: susceptible (S), vaccinated (V), infected (I), and recovered (R). The differential equations governing the dynamics are:

$$\begin{aligned}
\frac{dS}{dt} &= mN - \beta\frac{SI}{N} + \gamma R - vS - \mu S, \\
\frac{dV}{dt} &= vS - \xi\frac{VI}{N} - \mu V, \\
\frac{dI}{dt} &= \beta\frac{SI}{N} + \xi\frac{VI}{N} - rI - \mu I, \\
\frac{dR}{dt} &= rI - \gamma R - \mu R.
\end{aligned}$$

Here, N represents the total population, m the birth rate, μ the death rate, β the infection rate, γ the rate of immunity loss, v the vaccination rate, r the natural recovery rate, and ξ the infection rate among vaccinated individuals. For the scenario in Sao Paulo, Brazil, during the 2009 H1N1 outbreak, the parameters are set as follows: $r = 1.17$, $N = 42{,}075{,}716$, $\gamma = 0.003$, $\mu = 0.00025$, and $m = 0.0006298$. Initially, it is assumed that there is only one infected individual. This model aids in predicting the spread and control of the disease by adjusting parameters like β, ξ, and v to simulate different outbreak scenarios and intervention strategies.

Midpoint Method

The Midpoint Method is a numerical technique used to approximate solutions to differential equations. It is particularly useful for systems where direct analytical solutions are challenging. In this method, the update from a point y_n at time t_n to y_{n+1} at time t_{n+1} involves the following steps:

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \\ y_{n+1} &= y_n + hk_2, \end{aligned}$$

where h is the step size, and $f(t,y)$ is the function defining the differential equation $\frac{dy}{dt} = f(t,y)$. This method estimates the midpoint of the interval for a better approximation of the next value, y_{n+1}.

Modified Euler Method

The Modified Euler Method, also known as Heun's Method, is a numerical technique to solve differential equations. This method improves the accuracy of the Euler method by incorporating an additional step to refine the estimate of the next value. The process for updating from point y_n at time t_n to y_{n+1} at time t_{n+1} is as follows:

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f(t_n + h, y_n + hk_1), \\ y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2), \end{aligned}$$

where h is the step size, and $f(t,y)$ is the function that defines the differential equation $\frac{dy}{dt} = f(t,y)$. The method first predicts an endpoint using Euler's basic step and then takes the average of this and the initial slope to estimate the next point, improving the accuracy of the approximation.

Runge Kutta Order 4 Method

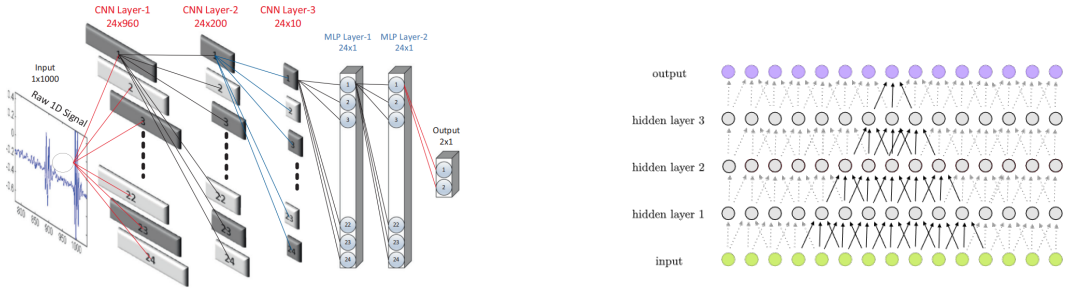
The Runge-Kutta method of order 4 (RK4) is a technique for obtaining approximate solutions to differential equations. It provides a good balance between complexity and accuracy. The method updates values using four intermediate calculations (k-values), which help predict the future behavior of the system with a high degree of accuracy. The process for updating from y_n at time t_n to y_{n+1} at time t_{n+1} is defined by:

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right), \\ k_4 &= f(t_n + h, y_n + hk_3), \\ y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \end{aligned}$$

where h is the step size, and $f(t,y)$ is the function that defines the differential equation $\frac{dy}{dt} = f(t,y)$. Each step uses information from the previous point to make a more informed prediction about the next point, thus improving the accuracy of the approximation significantly.

III. Machine Learning Systems

III.1 1D Convolutional Neural Network



(a) 1D CNN – Model Architecture

(b) 1D CNN – Training Results

Figure 1: 1D Convolutional Neural Network

The One-Dimensional Convolutional Neural Network (1D CNN) [2] is primarily used for analyzing sequences, such as time-series data. This network applies convolutional layers along the sequence, extracting features by sliding filters (kernels) across the sequence. The main operations in a 1D CNN are:

- Convolution: Applying filters to a sequence to produce feature maps.
- Activation: Introducing non-linearity into the feature maps.
- Pooling (optional): Reducing the dimensionality of each feature map.

The convolution operation in a 1D CNN can be described mathematically as follows:

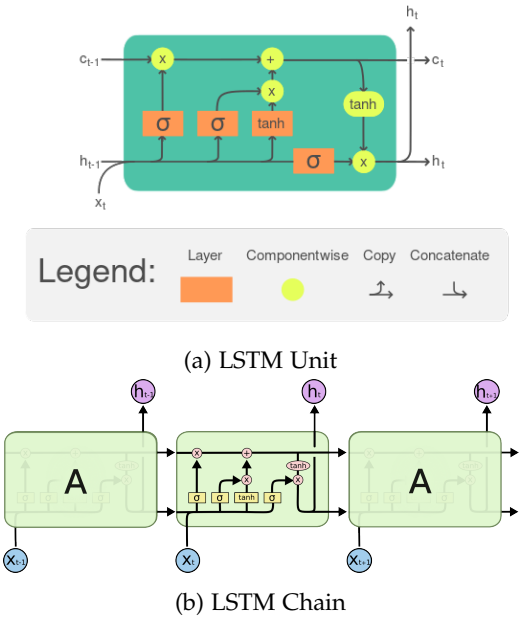
$$c_i = f\left(\sum_{k=0}^{K-1} x_{i+k} w_k + b\right),$$

where:

- x_{i+k} is the input at position $i+k$.
- w_k is the weight of the filter at position k.
- b is a bias term.
- f is a non-linear activation function, such as ReLU.
- c_i is the output of the convolution at position i.
- K is the size of the filter.

This formula indicates how each segment of the input sequence is transformed through linear combinations of inputs and filter weights, followed by a non-linear activation, to produce outputs that capture local dependencies in the data.

III.2 Long-Short-Term-Memory (LSTM)



(a) LSTM Unit

(b) LSTM Chain

Figure 2: Structure of Long-Short-Term-Memory (LSTM)

The Long Short-Term Memory (LSTM) [1] network is a special kind of recurrent neural network designed to remember information for long durations of time. It achieves this through a system of gates that control the flow of information. The basic operations in an LSTM cell are outlined by the following equations [**?**]:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad \text{(forget gate)} \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad \text{(input gate)} \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \quad \text{(candidate values)} \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t, \quad \text{(cell state update)} \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad \text{(output gate)} \\
h_t &= o_t * \tanh(C_t), \quad \text{(new hidden state)}
\end{aligned}
$$

Here, x_t is the input at time step t, h_t is the hidden state, and C_t is the cell state. W and b are weights and biases for different gates, and σ denotes the sigmoid function. This structure helps LSTMs to avoid the long-term dependency problem typical in traditional RNNs by carefully managing information through its state and gates.

III.3 Feature-Temporal Attention-based LSTM (FTA-LSTM)

The FTA-LSTM [3] model integrates both feature and temporal attention mechanisms with a Long Short-Term Memory (LSTM) network to enhance predictive accuracy and interpretability in deep learning models, as shown in Fig-3. This hybrid structure allows the model to selectively focus on the most relevant features and temporal segments of the input data, improving the handling of complex spatiotemporal relationships.

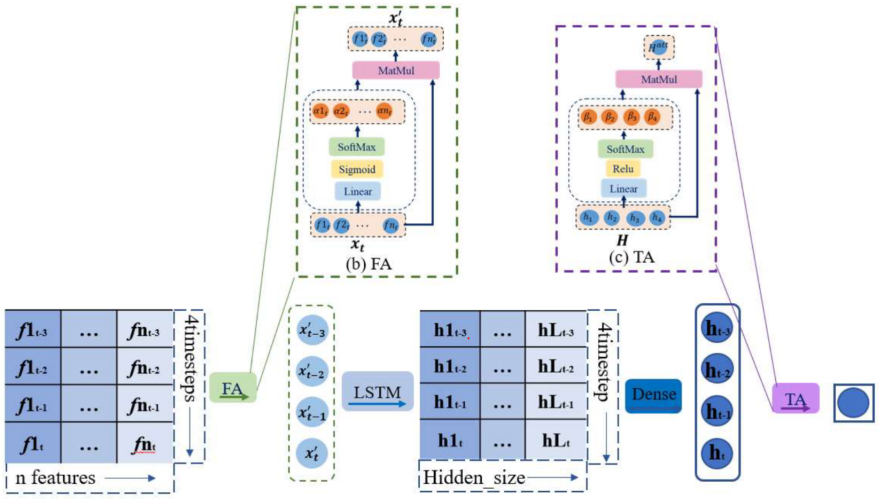


Figure 3: FTA-LSTM

Feature Attention in FTA-LSTM: The Feature Attention (FA) component of the FTA-LSTM applies a sigmoid-activated linear transformation to the input features, computing attention weights that scale the importance of each feature dynamically. The equations governing this process are:

$$\alpha_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{b}_f), \tag{1}$$

$$\mathbf{x}'_t = \alpha_t \odot \mathbf{x}_t, \tag{2}$$

where $\mathbf{x}_t$ is the input vector at time t, σ denotes the sigmoid function, $\mathbf{W}_f$ and $\mathbf{b}_f$ are the weight matrix and bias vector of the feature attention layer, respectively, and $\odot$ denotes element-wise multiplication.

LSTM Layer: Following feature attention preprocessing, the weighted input $\mathbf{x}'_t$ is processed by an LSTM layer, which updates its hidden state $\mathbf{h}_t$ and cell state $\mathbf{C}_t$ as follows:

$$(\mathbf{h}_t, \mathbf{C}_t) = \text{LSTM}(\mathbf{x}'_t, \mathbf{h}_{t-1}, \mathbf{C}_{t-1}), \tag{3}$$

where $\mathbf{h}_{t-1}$ and $\mathbf{C}_{t-1}$ are the previous hidden and cell states of the LSTM.

Temporal Attention in FTA-LSTM: Post-LSTM, the Temporal Attention (TA) component assigns weights to different time steps of the LSTM output, focusing on more relevant temporal segments. The temporal attention mechanism computes:

$$\beta_t = \text{Softmax}(\mathbf{W}_t \mathbf{h}_t + \mathbf{b}_t), \tag{4}$$

$$\mathbf{H}' = \sum_t \beta_t \mathbf{h}_t, \tag{5}$$

where $\mathbf{W}_t$ and $\mathbf{b}_t$ are the weights and biases of the temporal attention layer, and Softmax normalizes the attention scores across time steps.

Output Layer: The final output of the FTA-LSTM model is obtained by passing the temporally weighted hidden states $\mathbf{H}'$ through a fully connected layer:

$$y = \mathbf{W}_o \mathbf{H}' + b_o, \tag{6}$$

where $\mathbf{W}_o$ and b_o are the weights and bias of the output layer.

The combination of feature and temporal attention mechanisms in the FTA-LSTM model allows for a nuanced understanding of the data's dynamics, enabling the model to make more accurate and interpretable predictions, particularly useful in time-series forecasting tasks such as predicting the prevalence of diseases like influenza.

IV. Random Forest

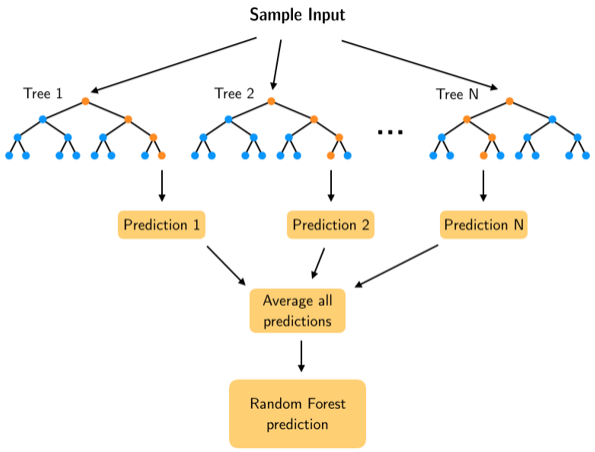


Figure 4: Random Forest

The Random Forest [4] model is a sophisticated ensemble learning technique that enhances predictive accuracy and reduces the likelihood of overfitting through the integration of multiple decision trees, as depicted in Fig-4. For our experiments, we have used 100 decision trees. Each tree in the Random Forest is constructed using a different subset of the data and features, which diversifies the model's predictions. The final output of the model is the average of all the individual trees' predictions, which is represented by the following equation:

$$y(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} T_i(\mathbf{x}), \tag{7}$$

where $T_i(\mathbf{x})$ is the prediction of the i-th tree, $\mathbf{x}$ represents the input features, and N is the number of trees in the forest.

In this study, the Random Forest model is utilized to perform regression analysis on historical influenza data, aiming to predict future case numbers based on past trends. We configure the model to use past data points (weekly case counts) as input features to predict the number of cases in future weeks. This regression task leverages the Random Forest's capability to handle non-linear relationships and interactions among features, making it particularly suitable for the complexities associated with epidemiological data, which often exhibits seasonal and non-linear patterns.

V. Evaluation Metric

To assess the performance of the numerical and machine learning models used in our study, we employ the Mean Absolute Error (MAE) as the primary evaluation metric. MAE is a widely used measure in regression analysis that quantifies the average magnitude of the errors between predicted values and observed true values, providing a straightforward interpretation of model accuracy. It is defined by the following equation:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{8}$$

where n is the number of samples, y_i are the true values, and $\hat{y}_i$ are the predicted values from the model. MAE is particularly useful in our context as it directly reflects the average error in the predicted number of influenza cases, allowing us to compare the effectiveness of different modeling approaches under the same criterion.

IV. Results

I. Evaluation of Numerical Methods

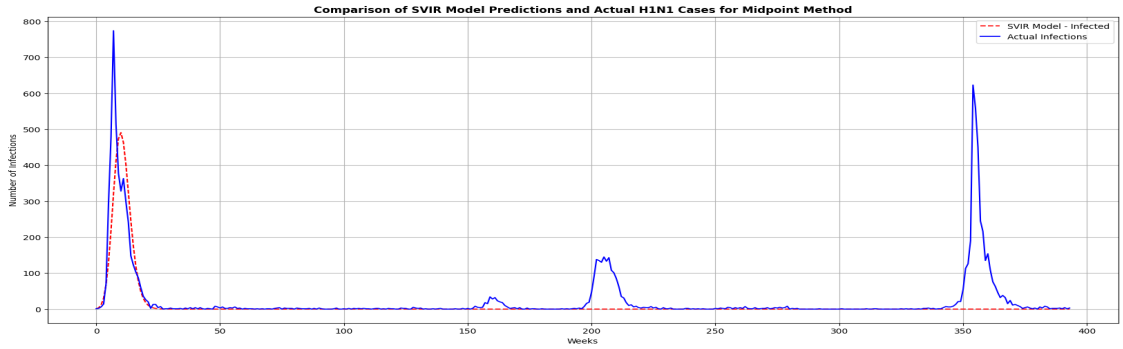
In our assessment of numerical methods for predicting influenza, we compared the Midpoint, Modified Euler, and Runge-Kutta Order 4 (RK4) methods. These methods were evaluated based on their Mean Absolute Error (MAE) values during both the training and testing phases, as summarized in Table 1. The results reveal similar performance metrics among the methods, with training MAEs ranging from 11.56 to 12.25 and testing MAEs from 44.31 to 44.35.

The comparative analysis, depicted in Figure 5, illustrates that all three numerical methods could approximate the initial peak of the influenza outbreak accurately when the time t (in weeks) is small. However, as time progressed, none of the numerical models effectively predicted the second peak of the outbreak. This suggests that while these methods provide a consistent level of accuracy for shorter forecast horizons, their precision diminishes over longer periods, particularly in the face of increasing variability and unpredictability inherent in the spread of infectious diseases.

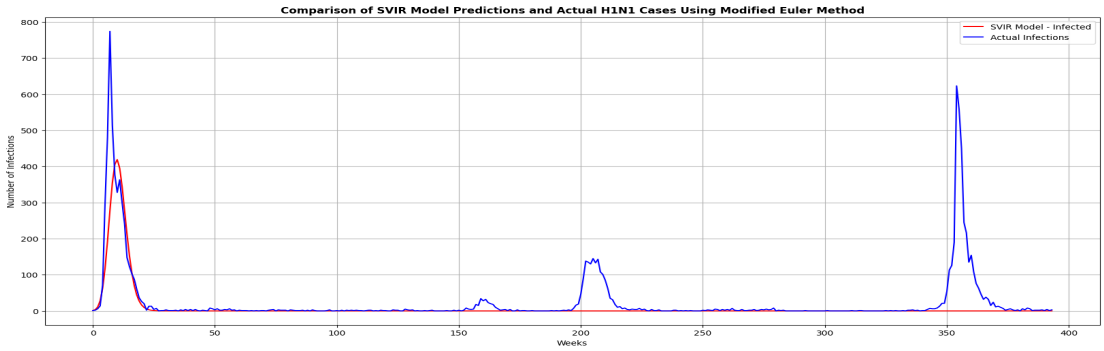
Interestingly, despite the RK4 method involving more computation due to its higher-order approximation, the results were similar to those obtained from the Midpoint and Modified Euler methods. This observation underscores the complex nature of epidemiological modeling, where increasing the complexity of the numerical method does not necessarily translate to significantly improved predictions, highlighting the limitations of using traditional numerical methods alone for long-term disease forecasting. Therefore, these findings suggest a potential integration of machine learning techniques to enhance the accuracy and reliability of predictions in public health scenarios.

II. Evaluation of Machine Learning Models

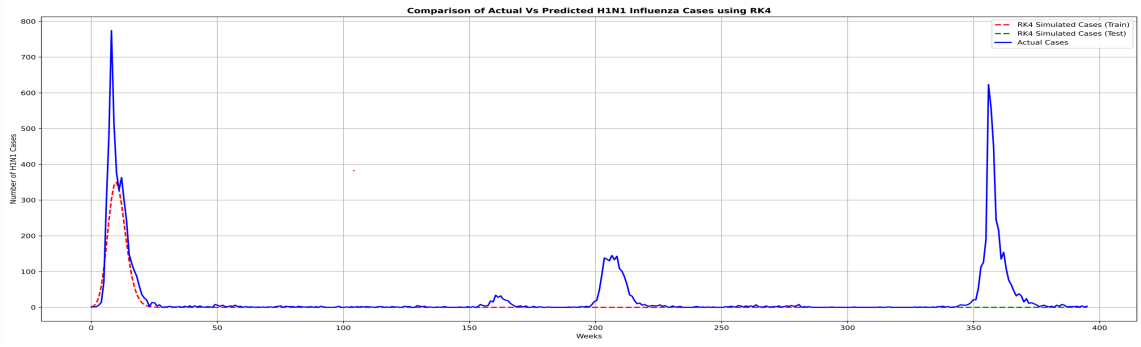
This subsection assesses the performance of various machine learning models in forecasting influenza cases using both direct and rolling forecast approaches. The effectiveness of these models is compared based on their Mean Absolute Error (MAE) metrics, detailed in Table 1, and visually represented in the plots under Figures 6 and 7.



(a) Modified Euler Method

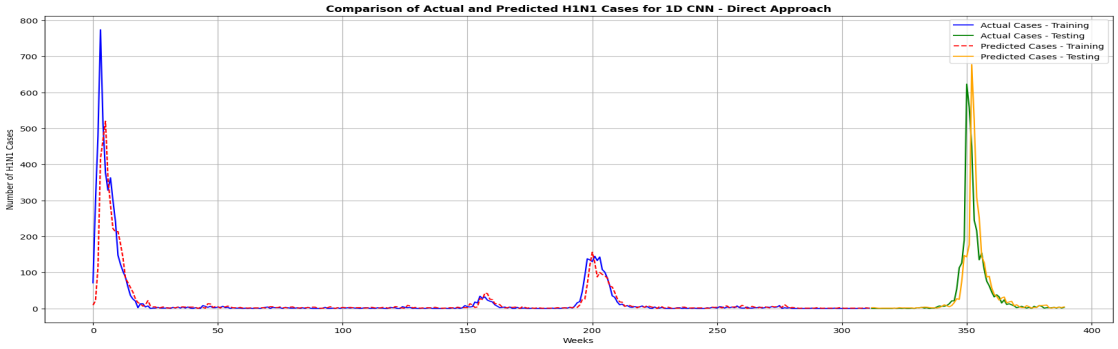


(b) MidPoint Method

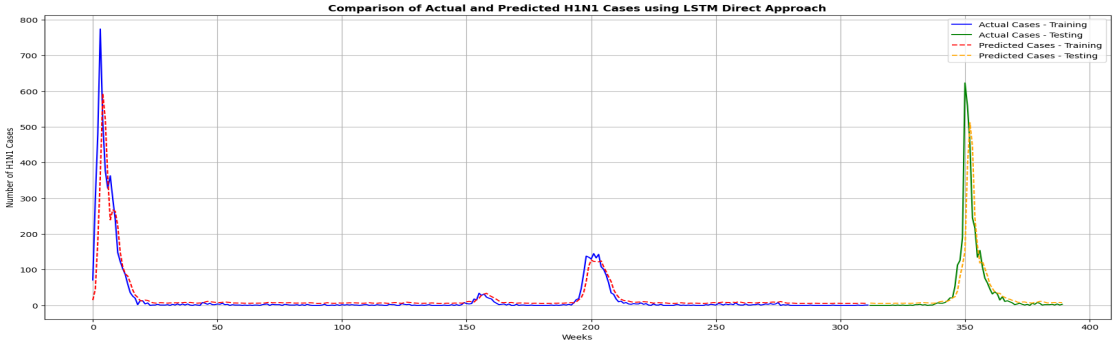


(c) Range-Kutta Order 4 Method

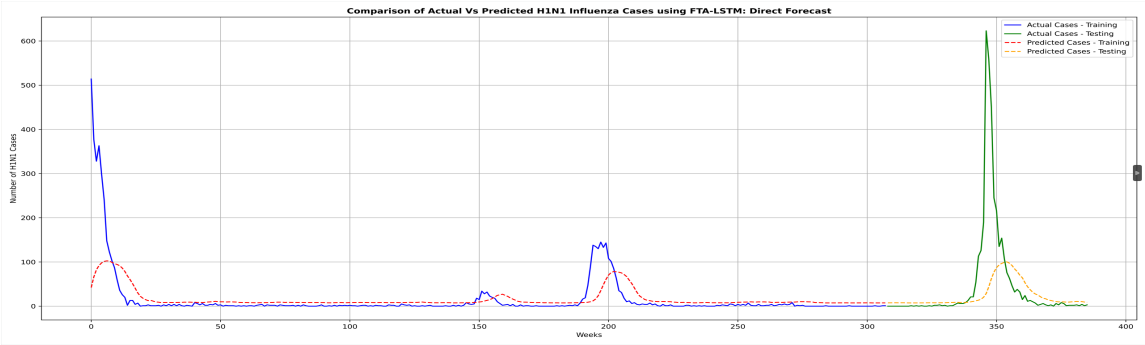
Figure 5: Evaluation of Numerical Methods for Influenza Prediction



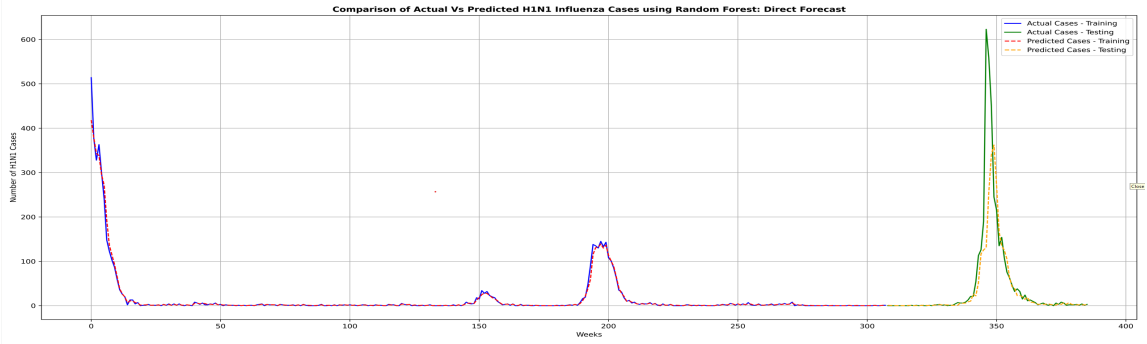
(a) 1D-CNN - Direct Forecast



(b) LSTM - Direct Forecast



(c) FTA-LSTM - Direct Forecast



(d) RF - Direct Forecast

Figure 6: Evaluation of Direct Forecast Approaches for Influenza Prediction

Category	Model	MAE	
		Training	Testing
Numerical Methods	Midpoint	12.25	44.35
	Modified Euler	11.56	44.31
	RK4	11.67	44.35
ML & DL Methods	1D CNN – Direct	10.06	25.05
	LSTM – Direct	11.35	20.49
	FTA-LSTM - Direct	10.26	24.16
	Random Forest - Direct	**2.09**	**14.67**
	1D CNN – Rolling	11.19	46.88
	LSTM – Rolling	6.97	44.42
	FTA-LSTM - Rolling	12.76	**44.39**
	Random Forest – Rolling	**2.11**	47.28

Table 1: Comparison of Numerical and ML & DL Methods

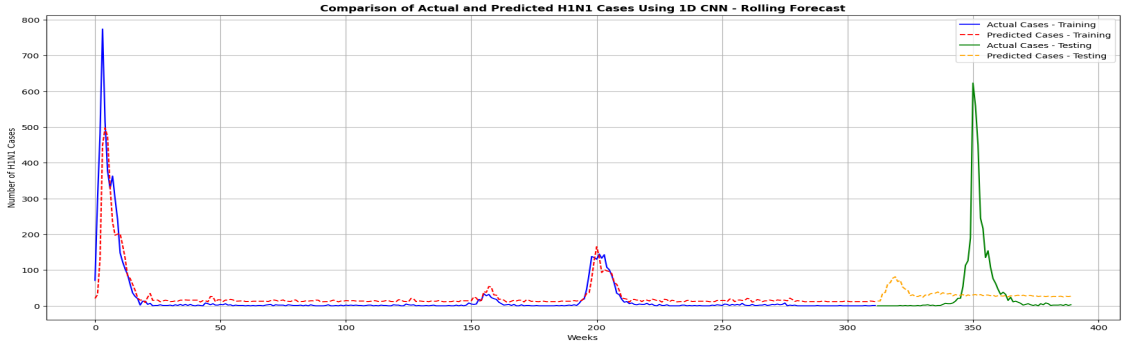
II.1 Direct Forecast Approach

A direct forecast approach involves using historical data to predict future values directly at once. Specifically, each input sequence contains ground truth past values in order to predict the next value. As shown in Figure 6, all machine learning models successfully approximated not only the training cases but also the testing set cases. This includes the minor peaks between the first major peak and the second peak. The 1D-CNN, despite slightly overshooting, provided the closest approximation to the second peak, followed by LSTM. The FTA-LSTM model exhibited the least accurate approximation of the second peak, with Random Forest delivering intermediate performance. However, in overall test set performance, the Random Forest model outperformed all other models in both training and test datasets, demonstrating a robust capability to generalize from the training data to unseen data. This is anticipated in the direct forecast approach since models are specifically trained to optimize performance over such structured data sequences.

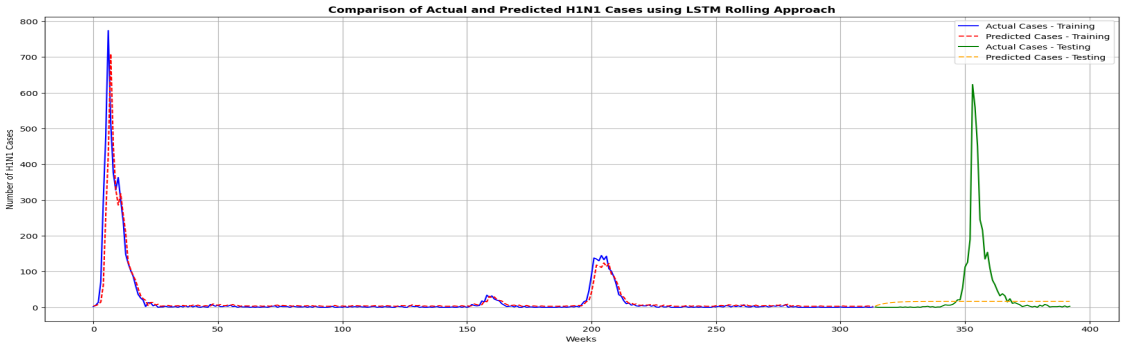
II.2 Rolling Forecast Approach

The rolling forecast approach entails using model outputs as the input for subsequent predictions, recursively over time. Specifically, the model uses its own approximated past values to predict the next value. Figure 7 illustrates that although all models managed to approximate the training data well, including the first peak and minor peaks, they failed to accurately predict the second peak in the test set. Instead, a flat line was observed for all models, indicating a divergence from the actual trend once the model predictions deviated from reality. This phenomenon is inherent to rolling forecasts; once the initial predictions err, all subsequent forecasts tend to propagate that error, leading to significant deviations in long-term predictions.

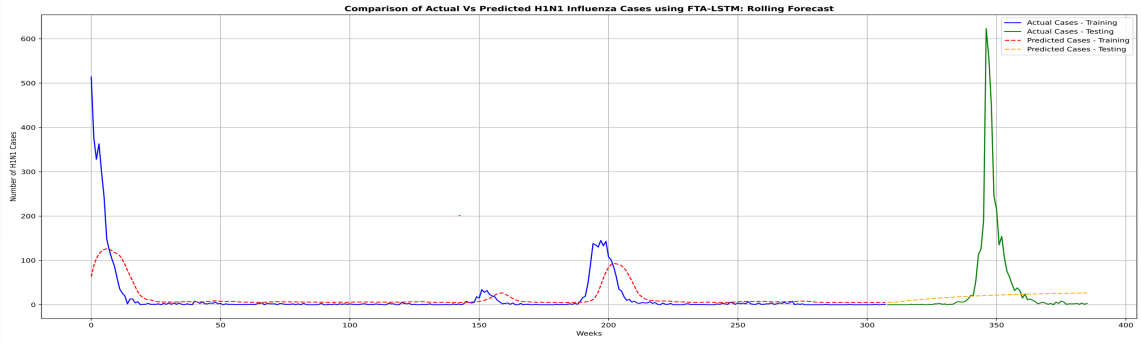
Both approaches highlight the strengths and limitations of machine learning models in handling epidemiological data. While machine learning models are capable of capturing complex patterns and dynamics in the data, their performance can vary significantly based on the forecasting strategy and the nature of the data they are trained on. The direct forecast models are generally more accurate and reliable when the training data encompass all variations expected in the test set, whereas rolling forecasts are more susceptible to cumulative prediction errors.



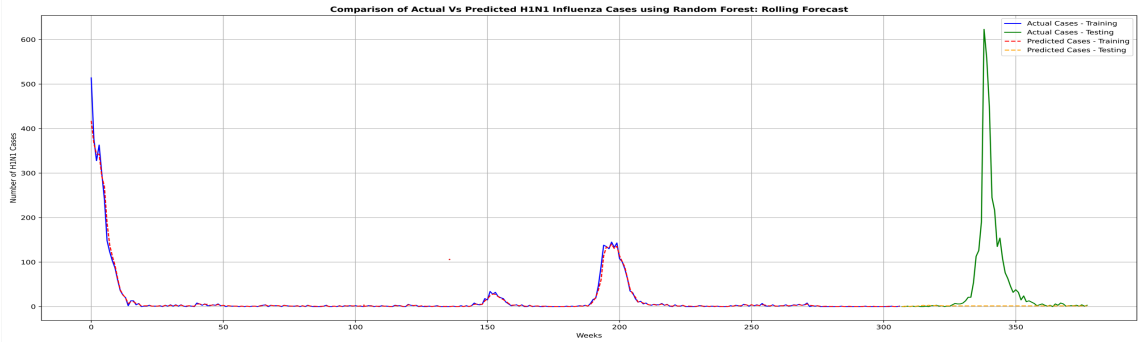
(a) 1D-CNN - Rolling Forecast



(b) LSTM - Rolling Forecast



(c) FTA-LSTM - Rolling Forecast



(d) RF - Rolling Forecast

Figure 7: Evaluation of Rolling Forecast Approaches for Influenza Prediction

III. Effect of Window Size in Time-Series Forecasting for ML Models

The variation in window size has a distinct impact on the Mean Absolute Error (MAE) of both Direct and Rolling Forecast models, as shown in Table 2. In the Direct Forecast approach, smaller window sizes generally yield better accuracy for most models, with Random Forest showing optimal performance at a window size of 4 (w=4). This suggests that a moderate window size, which balances recent and slightly older data, can be optimal for capturing relevant patterns without introducing noise from older data points. Conversely, larger window sizes appear to deteriorate performance, particularly for the FTA-LSTM, indicating that too much historical data can overwhelm the model, leading to poorer generalization.

Approach	**Models**	**MAE**		
		w=1	w=4	w=8
Direct Forecast	1D CNN	16.63	25.05	26.70
	LSTM	26.39	19.96	23.53
	FTA-LSTM	**16.48**	34.66	40.27
	Random Forest	20.35	**16.35**	**19.16**
Rolling Forecast	1D CNN	46.50	46.88	46.35
	LSTM	45.08	**45.22**	**44.81**
	FTA-LSTM	46.56	47.74	47.82
	Random Forest	**44.20**	46.22	48.33

Table 2: MAE Comparisons for Direct and Rolling Forecast Models

In Rolling Forecast models, the window size shows a less pronounced effect on performance, with all models demonstrating relatively stable MAE across different window sizes. This stability suggests that Rolling Forecast models might be less sensitive to the window size due to their reliance on previous forecasts rather than direct historical inputs, potentially leading to a consistent but less adaptive forecasting behavior across different scenarios.

IV. Numerical Methods Vs. Machine Learning Systems: A Comparative Approach

As demonstrated in Table 1, machine learning (ML) systems significantly outperform traditional numerical methods, particularly in direct forecasting scenarios. Notably, the direct forecast approach of ML models, such as the Random Forest, shows lower Mean Absolute Error (MAE) values, highlighting their superior accuracy in leveraging actual ground truth past values to predict future events Figure (6). Conversely, both rolling forecast ML methods and numerical methods exhibit limitations, particularly in predicting dynamic changes such as the second peak in the testing set. This is illustrated in Figures 7 and 5, where both approaches yield a flat prediction line during the second peak.

The similarity in the flat-line response during the second peak for both rolling forecast ML models and numerical methods stems from their reliance on previously generated data points for future predictions. This methodological constraint restricts their ability to adapt quickly to sudden changes in the data trend, which is critical in scenarios like epidemic forecasting. On the other hand, direct forecast ML methods, which utilize actual observed values from past data, are better equipped to handle such variability, resulting in more accurate predictions.

This difference underscores a fundamental distinction between the two approaches: numerical methods follow a deterministic model based on predefined mathematical formulations, which can be less flexible when encountering new or variable data patterns. In contrast, ML models, especially those employing a direct forecast strategy, adaptively learn from data, thus enhancing their ability to generalize from historical patterns to future predictions. This adaptability is a key factor in their lower error rates and more reliable performance in practical applications.

V. Conclusion

In this study, we have thoroughly analyzed and compared machine learning methods with traditional numerical SVIR models for predicting influenza trends. Our findings, detailed in our tables and figures, demonstrate that the Random Forest model is particularly effective, providing the most accurate forecasts. A key insight from our analysis is that the Direct Forecast Approach outperforms the Rolling Forecast Approach. This superior performance is attributed to the use of actual past values as inputs in the Direct Forecast Method, which enables more accurate predictions by using true historical data. In contrast, the Rolling Forecast Approach relies on its own previous forecasts as inputs, which can perpetuate initial inaccuracies. This distinction underscores why machine learning techniques, especially those employing a Direct Forecast strategy like Random Forest, significantly enhance forecasting accuracy compared to conventional numerical models.

References

[1] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.

[2] Tang, W., Long, G., Liu, L., Zhou, T., Jiang, J., and Blumenstein, M., 2020. Rethinking 1d-cnn for time series classification: A stronger baseline. arXiv preprint arXiv:2002.10061, pp.1-7.

[3] Wang, Y., Shi, L., Hu, Y., Hu, X., Song, W., and Wang, L., 2023. A comprehensive study of deep learning for soil moisture prediction. Hydrol. Earth Syst. Sci. Discuss

[4] Ho, Tin Kam. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, volume 1, pages 278–282. IEEE, 1995.