EWNETs and Signal Decomposition

Amey Choudhary, Aryan Gupta, Keshav Gupta

(Team Baby Sharks)

Dynamical Processes in Complex Networks, Spring 2024

Dr. Chittaranjan Hens

April 21, 2024

Abstract

In our project, we aim to understand the novel approach for forecasting epidemics, EWNETs, as proposed in Epicasting: An Ensemble Wavelet Neural Network for forecasting epidemics(2023) . We reproduce the same and analyze the results for it. We also experiment with other methods for signal decomposition and provide a comparative report for them.

Table of Contents

Understanding of the Paper

Epidemics are widespread occurrences of infectious illnesses. As they are among the top contributors of illnesses and death worldwide, their forecasting is an essential field of research. By predicting them accurately, it can assist stakeholders in developing countermeasures to contain and minimize their spread. This paper explains why epidemic forecasting is essential and how the current methods fall short, proposes a new method called as "EWNETs" , provides a mathematical base for their properties, and predicts various epidemics using EWNETs. The following subsections provide necessary details regarding the paper.

Current methods for epidemic forecasting

Epidemiological modeling is a centuries old field of research. Various methods have been developed over time for accurate forecasting. Several **mechanistic (or deterministic)** and **phenomenological** methods have been proposed.

Compartmental models, a subset of deterministic models, are widely used to study characteristic changes and state of population, by dividing it into "compartments". Models like SIR, which use simple differential equations to model the epidemic growth, are also used to provide insights into disease epidemics. In spite of their applicability, they tend to explain the disease dynamics better than predicting the spread.

To overcome this, statistical and machine learning methods have been introduced. While statistical methods rely on parametric methods on data, machine learning methods have been data-driven. Few examples of statistical and machine learning methods for epidemic forecasting are:

- Statistical
    - Random Walk
    - Autoregressive Integrated Moving Average (ARIMA) & its variations
    - Bayesian Structural Time Series
    - Exponential Smoothing State Space
- Machine Learning
    - Artificial Neural Networks
    - Support Vector Regression
    - RNN and LSTM
    - Transformers
    - Hybrid ARIMA-ANN

These models take input of time series data and output the predicted number. Several of these models, however, need to make assumptions and impose restrictions on the dataset for their application. Real-world epidemic datasets are **complex, noisy, non-stationary and non-linear systems,** and so, if inputted directly into our models, shall not be able to give accurate outputs. This brings out the need for preprocessing data in epidemic forecasting. By using mathematical transformations on our original data, we can obtain more accurate results. Log transformations and Fourier transformations, both are popular tools. While the former conforms the original data to normal distribution, the latter is used for periodic signals. These transformations change the symmetric error measurements on the original scale to asymmetric measures on log scale, are impacted by outliers (log transformations) and not suitable for non-periodic signals.

To overcome these problems, another kind of transformations, called wavelet transformations, are used. Wavelet transformations are used to decompose original data series into simpler high and low frequency coefficients, which allow us to differentiate signal from the noise. A common wavelet transform is **Discrete Wavelet Transform (DWT)**, which is used to decompose original signal to constituent signals. DWT, however, imposes restriction on signal length. To avoid this, Maximum Overlap DWT is used. Various new models involve signal decompositions, followed by a statistical or machine learning approach on the decompositions. Individual predictions are then recomposed to get overall future predictions. Some examples of wavelet based approaches are ARIMA-WARIMA, Wavelet-NBeats, Wavelet-Transformers.

These models, however, require huge data points for learning. Also, these models lack desired theoretical properties. The paper thus introduces a new model, EWNETs, and provides more accurate results for short, medium and long term forecasts. It also provides the mathematical basis behind EWNETs.

Explaining EWNETs

We shall now explain the proposed model, EWNETs. It consists of signal decomposition using MODWT and then using autoregressive neural networks to predict ahead.

MODWT approach

To understand the Maximal Overlap Discrete Wavelet Transform (MODWT), it's helpful to contrast it with the more traditional Discrete Wavelet Transform (DWT) and explore why MODWT was developed.

Discrete Wavelet Transform (DWT):

The DWT is a widely used method for analyzing signals and time series data. It decomposes a signal into different frequency components using wavelet basis functions. However, traditional DWT suffers from a few limitations:

1. Boundary Effects: When applying DWT to signals with finite length, boundary effects can occur due to discontinuities at the edges of the signal. This can distort the wavelet coefficients near the boundaries, leading to inaccurate representations.

2. Lack of Shift Invariance: DWT is not shift-invariant, meaning that small shifts in the signal can result in significant changes in the wavelet coefficients. This lack of shift invariance can be problematic in applications where robustness to shifts in the signal is required.

Maximal Overlap Discrete Wavelet Transform (MODWT):

To address the limitations of DWT, the MODWT was developed. MODWT introduces several key concepts:

1. Circular Convolution: MODWT extends the signal using circular convolution before applying the wavelet transform. This circular extension effectively treats the signal as periodic, eliminating boundary effects that can arise in traditional DWT.

2. Maximal Overlap: MODWT maximizes the overlap between adjacent wavelet coefficients by extending the signal and applying the wavelet transform with a shift. This maximal overlap

ensures that the decomposition process captures as much information as possible from the signal, enhancing resolution and fidelity.

3. Iterative Decomposition: Similar to DWT, MODWT decomposes the signal into different frequency bands or scales. However, MODWT typically applies multiple levels of decomposition, each refining the representation of the signal and capturing finer details.

4. Update Formulas: After each level of decomposition, MODWT updates the signal using specific formulas that combine approximation and detail coefficients. These update formulas ensure that the updated signal preserves important signal characteristics while minimizing boundary effects.

EWNET model architecture

The paper proposes a novel model, "EWNETs". This model utilizes MODWT signal decomposition as a preprocessing step for data. MODWT additively decomposes the data into J detail levels and 1 smooth level (i.e. for any time point t, the sum of corresponding points in details and smooth shall yield back the original data point)

$$Y_t = \sum_{j=1}^{J} D_{j,t} + S_{J,t}.$$

(where Y_t denotes the original data point at time point t, $D_{j,t}$ indicates value of time point t in detail D_j, $S_{J,t}$

indicates value of time point t in smooth)

These J+1 components are then fed into an ensemble of autoregressive neural networks (J+1 neural networks are present in the ensemble; each neural network takes 1 component as input). The input to them is p-lagged, i.e. for every point to output, p previous time point values are given as input. Autoregressive means the output is used as input in the next time point.

The architecture of these neural networks is that they contain an input layer (of p neurons), a hidden layer (of k neurons) and an output layer (1 neuron). Activation function is sigmoid. The weights are initialized randomly and trained using gradient descent. As for p, it is a hyperparameter, whose choice is based on minimization of forecast error on validation set in a cross-validation way. The value of k is [(p+1)/2] (determined in statistical properties). After prediction, the outputs of these J+1 components are added to give final output.

$$\hat{D}_{j,N+1} = \alpha_{0,j} + \sum_{i=1}^{k} \beta_{i,j}\phi(\alpha_{i,j} + \beta'_{i,j}\underline{D}_j);\ j = 1, 2, \ldots, J,$$

$$\hat{S}_{J,N+1} = \eta_0 + \sum_{i=1}^{k} \delta_i\phi(\eta_i + \delta'_i\underline{S}_J);$$

where $\underline{D}_j$, $\underline{S}_J$ denotes p lagged observations of the corresponding decomposed series ($j = 1, 2, \ldots, J$), $\alpha_{0,j}$, η_0, $\alpha_{i,j}$, $\beta_{i,j}$, η_i, δ_i ($i = 1, 2, \ldots, k; j = 1, 2, \ldots, J$) are the connection weights of the network, $\beta'_{i,j}$, δ'_i are p dimensional weight vectors, and ϕ is the nonlinear activation function (precisely, logistic sigmoidal activation function). The weights of the network take random values (how the details and smooth for next time point are calculated)

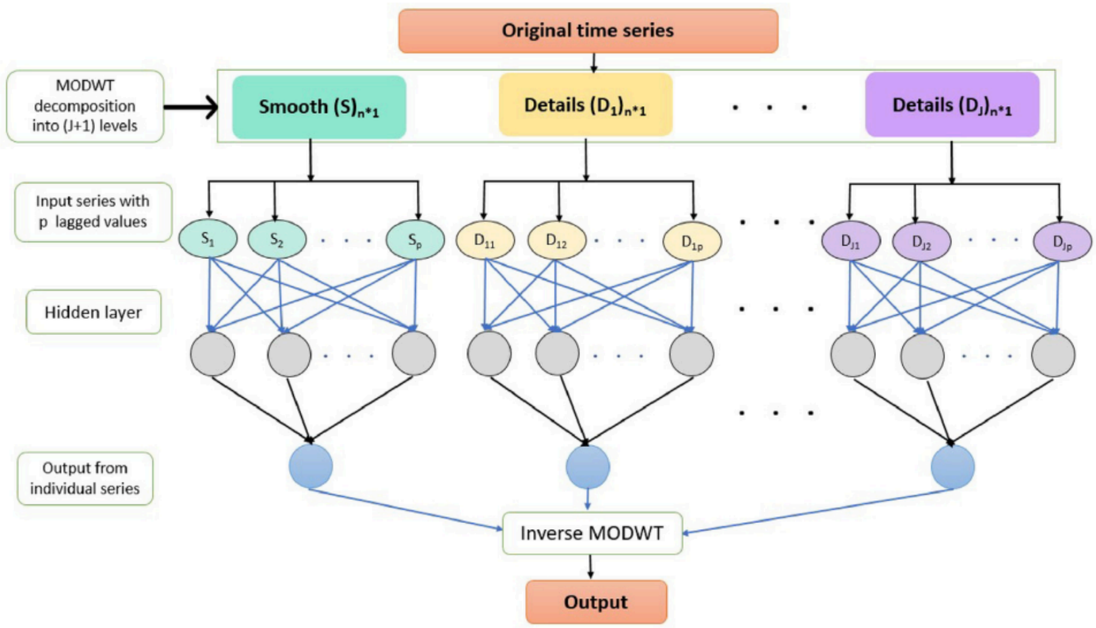
$$\hat{Y}_{N+h} = \sum_{j=1}^{J} \hat{D}_{j,N+h} + \hat{S}_{J,N+h}$$ (recomposition)

$$p = \underset{p}{argmin} \frac{1}{|V|} \sum_{t \in V} \frac{2|\hat{Y}_t - Y_t|}{|\hat{Y}_t| + |Y_t|} * 100\%,$$ (choosing p)

We have explained the architecture of the overall model. Other relevant hyper parameters include setting the number of detail coefficients (J) and setting the forecast horizon (length of test prediction)

We shall explain its working next.

EWNET model working



While Training:

1. The original time series data, Y (y(t) denotes value of time series data at time point t) is fed into the model first. Using MODWT, the time series is broken down into J+1 components (J details and 1 smooth component). The MODWT algorithm uses a pyramid algorithm.
2. Each component is transformed to a time series object. This component is fed into the neural network to model. So each component's time series (let c denote the time series of the component) corresponds to a particular neural network.

3. These neural networks are p-lagged, i.e. for every time point t, it takes the input as the set [c(t-1), c(t-2) …… , c(t-p)] and tries to model them.
4. We get the outputs and find the loss. Note that the neural networks are trained independently and so the loss is found out from the predicted value and the exact value in the corresponding component c (and not from Y). We change weights accordingly by gradient descent. We train all the neural networks for some error threshold / epoch threshold.

While Testing:

1. We are given a signal, I, consisting of N points, and we need to predict the next "h" points.
2. The signal is decomposed into various components and then passed onto the trained neural network.
3. In each neural network, as input should be p-lagged, the neural network takes the previous p values of the component signal. So in order to predict the first point of the output (o_1), it takes the last p values of the input component. As the model is autoregressive, it includes the output as input for the next point. So when we try to predict the second point (o_2), it takes

the last p-1 values of the input component and the recent output (o_1) as input. This goes on and the neural network includes its previous outputs as inputs for the next point (autoregressive nature) alongside remaining historical data. After p values are predicted (assuming p<h) the model depends entirely on its previous outputs.

4. After each neural network predicts h points (h x (J+1) points predicted in total), predictions for the same time point are added and the resultant sum is the final prediction for that time point.

Thus, the model has been trained and tested.

Experimentation results of EWNETs on datasets

The authors of the paper carry out experiments to test the accuracy of EWNETs and compare it with other common models for forecasting (statistical, machine learning, deep learning and hybrid). They use **15** real-world epidemic datasets for testing their model. To demonstrate its generalizability, they predict over various forecast horizons and analyze its epicasting performance for three different forecast horizons — **long, medium, and short-term** spanning over (**52, 26, 13**) weeks for weekly datasets and (**12, 6, 3**) months for monthly datasets, respectively. For benchmarking, they use **RMSE, MASE, MAE, sMAPE** as the criterion. The optimal value of “p” (lagging parameter) is determined by a grid search (minimum MASE score). Value of k is the floor value of [(p+1)/2].

The results of the model show that on average, our EWNET model is able to beat out other state of the art learning models. This phenomenon occurs due to

- the lack of a humongous amount of historical data in most datasets. Most deep learning methods are highly suitable for high-frequency datasets. However, these are seldom available, hence the applicability of these models is limited, especially in the epicasting domain
- the non-stationary and nonlinear characteristics of the real-world epidemic datasets. The wavelets coupled with ARNN in an ensemble framework (as done in the EWNet architecture) capture the non-stationary and seasonality of the time series using the wavelet decomposition, whereas the ARNN is responsible for handling nonlinear behavior. (wavelets are able to capture the non-stationarity and seasonality, while ARNN captures nonlinearity.
- the epidemic data sets exhibit long range dependency, which the ARNN framework present in the forecasting stage of the EWNet model can generate more reliably.
- the performance of the multi-head Transformers model is significantly worse than the majority of the forecasters. This is because Transformers can

accurately extract semantic relations among the elements in a long sequence, in a time series modeling for extracting temporal correlations in an ordered sequence, the model employs positional encoding and tokenizes the dataset into several sub-series. This nature of the permutation-invariant self-attention mechanism eventually leads to the loss of temporal information resulting in imprecise forecast

- the wavelet-based deep learners W-Transformers and W-NBeats lack the desired theoretical basis that restricts the model from showing ‘explosive’ behavior or growing variance over time, and are so unreliable

The EWNET model can thus be used in epidemic forecasting, as an early warning system. The theoretical basis for selecting the model’s hyperparameters significantly reduces its run-time complexity compared to state-of-the-art deep learners, which enables real-time real-time forecasts.

Additionally, several other external factors, can be employed (like temperature, rainfall, geographical scale) to increase the accuracy

Implementing our own EWNET

We have implemented our EWNET in the following manner:

1. Data preprocessing

   We read the data from the file. On the basis of the type of dataset, we choose "Weekly" or "Monthly". The type of forecast horizon is also chosen. Hyperparameter values are taken from the excel containing optimal values. We also scale the data from 1 to -1, in order to ensure neural networks are able to model them well.

2. MODWT algorithm

   The MODWT algorithm, with Haar filter, is used to decompose the signal. The number of levels depends upon the floor value of the natural logarithm of the number of data points in training. The MODWT algorithm decomposes the signal into a number of details and smooth coefficients.

3. Training of Ensemble of Neural Networks

   We use the details and smooth coefficients , in training the neural network. Our neural network contains p input neurons, k hidden neurons and 1 output neuron.The activation function is sigmoid.

   For each neural network, we select a corresponding coefficient on which it will be trained. We feed a lagged sequence of the corresponding coefficient , as input to our neural network. Our neural network is trained to predict the next term of the sequence for its corresponding coefficient.

4. Testing and recomposition

   To predict, we first feed in the last p values of the coefficients to the corresponding neural network. We do one-step ahead forecasting and get the next prediction. This next prediction is appended at the end of test input. As we again predict the next sequence, we take the last p values (last value is the recent output and p-1 values from before). We again append the output. This continues and establishes the auto-regressive nature of the neural network (previous outputs act as input for next sequence). Once we get predicted future sequence of all coefficients, we recompose using an inverse function (additively recomposes). The final output for a time point is the sum of predicted values of the time point in all coefficients.

Codes for our implementation of EWNET and other signal decomposition techniques can be found here: DPCN_Baby_Sharks_github

Using other Signal Decomposition Methods

We also tried various signal decomposition methods in place of MODWT. The change comes in signal decomposition. Ensemble of neural networks remain the same.

1. Seasonal and Trend decomposition using Loess (STL)

Seasonal and Trend decomposition using LOESS (locally estimated scatterplot smoothing), is a signal decomposition method, which decomposes the signal into seasonal, trend and residue. Here trend means the underlying long term progression or direction in data, seasonal means the repeating patterns in a period (days, weeks, months or years) and residual is the arbitrary noise or fluctuations which cannot be fitted. LOESS is a non-parametric method which uses linear regression in a localized subset of data, to build up a function, which can model that subset. LOESS algorithm first approaches each point in the data, and fits a linear local polynomial over a subset of data (length of subset depends upon the bandwidth or smoothing parameter). A weighted function is used to assign weights to points in a data subset, with nearby having more weight than far off. Any new input points output is determined by the weighted local regression, where weights are determined by the distance of nearby points.

STL uses the LOESS algorithm to decompose. It first takes the input signal, performs LOESS, and then, by using the local regression functions, it predicts the output for each input point. These output points are the trend signal. It then subtracts the trend from the input to get a new signal (detrended data). STL again employs LOESS on this detrended data to get the seasonal in the same manner as above. It removes the seasonal from the detrended data to get residue.

STL is advantageous as it is robust to outliers, can model any seasonality (daily, monthly , yearly .etc.) and can perform automatic parameter selection (for smoothing).

2. Singular Spectral Analysis (SSA)

SSA is a powerful technique which is applicable to study of classical time series, multivariate statistics, multivariate geometry, dynamical systems and signal processing. It does additive decomposition. SSA derives its name from the eigenvalues of the singular value decomposition of a covariance matrix. The algorithm for SSA in brief is:

1. Given the observed points series, Y, select a window length (L) , where L must be less than half of the total number of data points in Y (N). Let K = N - L +1.

2. Form the trajectory matrix, X, of dimensions L x K. The i^{th} row contains K sequential data points, beginning from ith data point in Y.

$$\mathbf{X} = (x_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} y_1 & y_2 & y_3 & \cdots & y_K \\ y_2 & y_3 & y_4 & \cdots & y_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & y_{L+2} & \cdots & y_T \end{pmatrix}.$$

3. Compute $S = XX^T$. Perform SVD on S. Let the eigenvalues of S be $\lambda 1,\ldots,\lambda L$, taken in the decreasing order of magnitude ($\lambda 1 \geq \ldots \geq \lambda L \geq 0$) and by $U1,\ldots,UL$ the orthonormal system of the eigenvectors of the matrix S corresponding to these eigenvalues. Now, if $V_i = X^T U_i/\sqrt{\lambda_i}$, then SVD of X can be written as

$$\mathbf{X} = \mathbf{X}_1 + \ldots + \mathbf{X}_d,$$

where

$$\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^{\mathrm{T}}$$

(d is the rank of X, usually L for real world data)

4. Now, based on the grouping parameter "l" , we partition the set of indices into l groups.

Let $I = \{i_1, \ldots, i_p\}$. Then the resultant matrix $\mathbf{X}_I$ corresponding to the group I is defined as $\mathbf{X}_I = \mathbf{X}_{i_1} + \ldots + \mathbf{X}_{i_p}$.

$$\mathbf{X} = \mathbf{X}_{I_1} + \ldots + \mathbf{X}_{I_m}.$$

5. Each X_{Ij} is hankelised then . Diagonal averaging is applied to each X_{Ik} to get reconstructed series for that group, $\tilde{Y}_k$.
6. All reconstructed series are summed, to get final reconstructed series.

The advantages of SSA are non-parametric and make no assumption of the data. It can work well with any kind of data (linear or nonlinear, stationary or non-stationary).

3. Empirical Mode Decomposition (EMD)

Empirical Mode Decomposition (EMD) is a technique used to break down complex data into simpler parts (just like separating the different instruments in a piece of music so we can hear each one clearly), these simpler parts are called Intrinsic Mode Functions (IMF), and each one represents a different rhythm or pattern in the data. There's a residue, which is the leftover trend after all the patterns have been removed.

EMD works as follows:

1. **Identification of Extremes**: Detects all local maxima and minima of the signal.

2. **Envelope Interpolation**: Draw smooth curves (envelopes) through the maximas and minimas.
3. **Mean Envelope Calculation**: EMD calculates the average of these two curves.
4. **Extraction of IMF**: The difference between the data and this average line is called an IMF. It shows one of the underlying patterns or cycles in the data.
5. **Repeat**: The remaining data (after subtracting the IMF) is treated as new data, and the process repeats. Each time, a new IMF is found, revealing another pattern.
6. **Stop**: Stop when the residue becomes a monotonic function, which cannot be decomposed further into meaningful IMFs.

EMD is a powerful tool for analyzing nonlinear and non-stationary signals (unpredictable and time dependent, eg. dynamic real-world phenomena), allowing the extraction of meaningful oscillatory modes embedded within complex data sets. It is commonly used in various fields such as economics, medicine, and geophysics for trend analysis, noise reduction, and feature extraction.

4. Hilbert Huang Transform (HHT)

The Hilbert-Huang Transform (HHT) is an advanced signal processing technique that combines Empirical Mode Decomposition (EMD) with the Hilbert Spectral Analysis to analyze non-linear and non-stationary signals. The process allows for the extraction of instantaneous frequency data.

Working Principle of HHT

1. Empirical Mode Decomposition (EMD):

- Decompose Signal: The first step is using EMD to break down the original complex signal into IMFs. Each IMF contains well-behaved oscillations that make them suitable for further analysis.

2. Hilbert Transform:

- Apply Hilbert Transform: Each of the IMFs from the EMD process is then subjected to the Hilbert Transform. This mathematical transformation helps in deriving the analytical signal from each IMF.

- Analytical Signal: The result of applying the Hilbert Transform is a complex-valued analytical signal for each IMF, where the real part is the original IMF, and the imaginary part is the Hilbert Transform of the IMF. -

Instantaneous Amplitude and Phase: From the analytical signal, we can calculate the instantaneous amplitude (envelope) and the instantaneous phase. The derivative of the instantaneous phase gives we the instantaneous frequency.

3. Hilbert Spectrum:

- Construct the Spectrum: By plotting the amplitude of each IMF against its instantaneous frequency over time, a Hilbert Spectrum is formed. This spectrum provides a time-frequency representation of the original signal.
- Energy and Frequency Over Time: The spectrum shows how the energy (amplitude) of various frequency components of the signal evolves over time.

Why Use HHT? -

Detailed Time-Frequency Analysis: Unlike Fourier transform, which only provides frequency information averaged over the entire signal length, HHT provides time-dependent frequency information, making it incredibly powerful for analyzing signals whose frequency components change over time.

-Handling Non-Linear and Non-Stationary Data: HHT does not require the signal to be linear or stationary. This makes it suitable for real-world signals that often do not adhere to these conditions, such as economic, seismic, or biomedical signals. By combining EMD and Hilbert Spectral Analysis, HHT is particularly effective in providing a clear and detailed view of how signals behave, change, and interact over time.

5. Empirical Wavelet Transform (EWT)

The Empirical Wavelet Transform (EWT) is a signal processing technique used for analyzing non-stationary signals, such as those encountered in audio, image, and biomedical data.

1. Signal Decomposition: EWT decomposes the input signal into different frequency components in an adaptive manner. Unlike traditional Fourier-based methods, EWT decomposes signals into components that are better localized in both time and frequency domains.

2. Filtering: In each iteration of the decomposition process, EWT applies a set of filters to the signal. These filters adapt to the local characteristics of the signal, capturing both its smooth and oscillatory components effectively.

3. Empirical Mode Decomposition (EMD): EWT uses a concept similar to Empirical Mode Decomposition, where the signal is decomposed into a series of oscillatory components called "Intrinsic Mode Functions" (IMFs). However, unlike EMD, which relies on sifting and interpolation, EWT employs iterative filtering.

4. Adaptive Frequency Detection: EWT adaptively detects the frequency content of the signal at each iteration. This adaptive frequency detection is crucial for capturing the time-varying nature of non-stationary signals.

5. Reconstruction: Once the signal has been decomposed into its empirical wavelets, it can be reconstructed by summing these components. The reconstructed signal typically retains the essential characteristics of the original signal while allowing for easier analysis and interpretation.

Experimentation Results

In this section, we describe results we obtained from our experiments of implementing EWNET, as well as other signal decomposition techniques.

I. MODWT

Datasets	Benchmarks	Short Horizon	Medium Horizon	Long Horizon
Ahemdabad Dengue	RMSE	17.18	17.76	10.76
	MASE	2.12	1.58	1.66
	MAE	14.52	14.32	8.67
	sMAPE	85.68	101.77	97.61
Austrailia Influenza	RMSE	29.38	14.53	68.00
	MASE	8.39	1.07	3.00
	MAE	29.38	13.36	59.79
	sMAPE	105.72	58.85	86.21
China Hepatitis	RMSE	6343.47	7536.50	8150.63
	MASE	0.78	1.33	0.18
	MAE	5817.66	6309.04	6154.68
	sMAPE	6.04	6.54	6.19
Venezuela Malaria	RMSE	282.94	491.29	331.58
	MASE	1.91	2.71	1.71
	MAE	250.37	461.20	285.18
	sMAPE	16.68	33.04	20.55

II. STL

Datasets	Benchmarks	Short Horizon	Medium Horizon	Long Horizon
Ahemdabad Dengue	RMSE	9.517241	15.320183	22.065373
	MASE	1.129099	1.301728	3.508521
	MAE	7.715512	11.767617	18.23055
	sMAPE	47.8988411	55.058149	117.944525
Austrailia Influenza	RMSE	69.543089	64.572884	67.328613
	MASE	3.009542	2.254125	3.315976
	MAE	60.692433	55.721969	52.665509
	sMAPE	83.661812	85.16189	105.918563
China Hepatitis	RMSE	7127.211012	11993.361574	15548.246719
	MASE	0.81101	2.330647	1.96209
	MAE	6030.672984	11014.639909	12265.735462
	sMAPE	6.272688	11.737288	13.618059
Venezuela Malaria	RMSE	448.583488	333.778688	569.344964
	MASE	2.696043	1.706886	3.089197
	MAE	353.181572	290.443776	514.321008
	sMAPE	25.041347	19.514405	41.336717

III. SSA

Datasets	Benchmarks	Short Horizon	Medium Horizon	Long Horizon
Ahemdabad Dengue	RMSE	8.890322	18.297535	21.628127
	MASE	7.848153	14.87175	15.222834
	MAE	1.14851	1.645105	2.929678
	sMAPE	51.999668	91.660593	132.177606
Austrailia Influenza	RMSE	43.263639	117.881616	86.480901
	MASE	35.876697	102.484144	60.9058
	MAE	1.77901	4.145799	3.83481
	sMAPE	56.759987	159.059448	120.155975
China Hepatitis	RMSE	6020.01388	4562.707589	12175.556356
	MASE	5337.195103	3506.979666	10934.604867
	MAE	0.717751	0.742061	1.749155
	sMAPE	5.531585	3.605742	10.563544
Venezuela Malaria	RMSE	198.667891	297.677802	363.580102
	MASE	178.105363	258.703245	313.763067
	MAE	1.359583	1.520353	1.884574
	sMAPE	11.339351	15.041557	22.873681

IV. EMD

Datasets	Benchmarks	Short Horizon	Medium Horizon	Long Horizon
Ahemdabad Dengue	RMSE	18.372612	21.151647	13.334628
	MASE	15.247914	17.922687	9.694146
	MAE	2.231402	1.982598	1.865666
	sMAPE	88.280225	108.180067	97.588262
Austrailia Influenza	RMSE	81.495975	93.384722	164.483573
	MASE	70.426865	79.733277	148.910826
	MAE	3.492241	3.225456	9.375867
	sMAPE	88.859336	73.349306	200.0
China Hepatitis	RMSE	7095.548391	13041.521952	10033.907396
	MASE	6003.715418	10341.25933	8390.884234
	MAE	0.807385	2.188163	1.342249
	sMAPE	6.195656	11.699416	8.771945
Venezuela Malaria	RMSE	341.482309	455.775946	404.415762
	MASE	259.746116	404.49535	350.410513
	MAE	1.982795	2.377147	2.104692
	sMAPE	16.642275	28.685599	25.982547

V. HHT

Datasets	Benchmarks	Short Horizon	Medium Horizon	Long Horizon
Ahemdabad Dengue	RMSE	6.447304	20.657797	27.85592
	MASE	5.191813	17.322969	21.542327
	MAE	0.759778	1.916258	4.145882
	sMAPE	52.825746	108.203419	114.57793
Austrailia Influenza	RMSE	78.740955	107.856326	83.034016
	MASE	69.076484	82.476189	68.319555
	MAE	3.42528	3.336415	4.301602
	sMAPE	89.401426	67.83592	122.329142
China Hepatitis	RMSE	10508.1634	5147.571276	15119.8742
	MASE	8498.41883	4631.385576	12002.0560
	MAE	1.142875	0.97998	17 1.91991
	sMAPE	8.615816	4.765578	12.980847
Venezuela Malaria	RMSE	273.470504	459.724233	357.948167
	MASE	233.50759	409.293385	309.063664
	MAE	1.782501	2.405344	1.856348
	sMAPE	15.302164	29.252895	22.437152

VI. EWT

Datasets	Benchmarks	Short Horizon	Medium Horizon	Long Horizon
Ahemdabad Dengue	RMSE	11.46	20.60	40.66
	MASE	1.17	2.34	6.44
	MAE	8.04	21.15	33.48
	sMAPE	49.46	148.44	140.79
Austrailia Influenza	RMSE	131.03	115.28	131.08
	MASE	5.66	3.89	7.20
	MAE	114.21	96.20	114.39
	sMAPE	103.67	160.77	137.76
China Hepatitis	RMSE	6262.45	6839.08	11550.55
	MASE	0.82	1.20	1.31
	MAE	6111.36	5675.93	8202.69
	sMAPE	6.45	5.70	8.99
Venezuela Malaria	RMSE	151.66	297.58	475.29
	MASE	0.94	1.41	2.54
	MAE	123.21	241.60	423.65
	sMAPE	7.79	16.04	32.59

Discussion and Conclusion

These tables compare the performance of different forecasting methods (MODWT, EWT, STL, SSA, EMD, HHT) across various datasets (Ahmedabad Dengue, Australia Influenza, China Hepatitis, Venezuela Malaria) and different forecast horizons (Short, Medium, Long).

MODWT

As speculated, MODWT performs quite good and gives competitive results on all the datasets. MODWT is a wavelet-based transform that decomposes a signal into different frequency components using a predefined set of wavelet functions. It operates in a similar manner to other wavelet transforms but specifically focuses on maximizing the overlap between adjacent subbands. The algorithm is also more focused and mindful when it comes to modeling the edges and taking care of edge effects, which can be detrimental to the overall performance. Also, the choice of the filters really has an effect on the performance. Since the choice of the filters was a ‘haar’ filter, we see that haar filters for these datasets was a good choice as it is able to capture the high frequency information really well.

EWT

EWT gives quite competitive performance compared to other approaches. This is predominantly due to the adaptive nature of the algorithm. EWT is a data-driven approach that adaptively decomposes a signal into components called empirical wavelets. Instead of using predefined wavelet functions, EWT derives wavelets directly from the signal itself, making it more adaptive to the signal's characteristics. This adaptability allows EWT to capture both smooth and oscillatory components of the signal effectively, making it suitable for analyzing non-stationary signals with varying frequency content.

EMD vs HHT

Comparing EMD (Empirical Mode Decomposition) and HHT (Hilbert-Huang Transform) in terms of their performance and intuition behind their effectiveness:

- EMD (Empirical Mode Decomposition):
  - Shows competitive performance in Short Horizons for some datasets like China Hepatitis but exhibits higher errors in Medium and Long Horizons across all datasets.
  - Generally provides moderate forecasting accuracy compared to other methods.

- Reasons: EMD decomposes the time series into Intrinsic Mode Functions (IMFs), which represent different oscillatory modes or components of the data.
  - Intuition: EMD is effective in capturing complex, non-linear, and non-stationary patterns in the data by adaptively decomposing it into simpler components. This adaptability allows EMD to handle various types of data with irregular patterns or trends. However, it may struggle with data containing strong noise or outliers, which can affect the accuracy of the decomposition and lead to less accurate forecasts, especially in longer horizons.

- HHT (Hilbert-Huang Transform):
  - Demonstrates relatively better performance in Short Horizons for certain datasets like Ahmedabad Dengue and Australia Influenza but shows higher errors in Medium and Long Horizons.
  - Performs competitively in Short Horizons but may not maintain the same level of accuracy in longer forecast horizons.
  - Reasons: HHT combines Empirical Mode Decomposition (EMD) with the Hilbert transform to analyze instantaneous frequencies and phases of the data.

- Intuition: HHT is particularly effective in capturing non-linear and non-stationary behaviors in the data. By decomposing the time series into IMFs and analyzing their instantaneous frequencies, HHT can capture time-varying patterns and trends more accurately. This makes it suitable for short-term forecasts where instantaneous changes in the data are crucial. However, similar to EMD, HHT may struggle with noisy data or long-term forecasts due to the adaptability of the decomposition process.

Both EMD and HHT excel in capturing non-linear and non-stationary patterns in the data, making them suitable for short-term forecasts where such characteristics dominate. However, their adaptability and sensitivity to noise may affect their accuracy in longer forecast horizons or in the presence of strong noise or outliers. Choosing between EMD and HHT depends on the specific characteristics of the dataset, the desired forecast horizon, and the level of noise present in the data.

STL

STl decomposes the data into seasonal, trend and residue. In general, the performance of STL is good, which can be attributed to it breaking down long term dependency into trend and short term dependency into seasonal, It is also robust to outliers, which allows it to perform better. STL performs comparatively well across all forecast horizons. A drawback of STL, in comparison to the methods can be that it decomposes only into 3 coefficients, thus causing some constituents to be grouped in 1, making it harder for neural network to model them

SSA

SSA decomposes the signal into various coefficients.It has comparable, if not better performance than others. It works exceptionally well in Short Horizon. As it is non-parametric, and it can decompose any signal into coefficients. As these coefficients are plenty, it is able to appropriately decompose data into simple enough constituents, which can be modeled easily by the neural network, thus attributing its performance. As it also uses eigenvalues to determine the most important components,, it is able to separate them out first and ensure that high weightage constituent signals are modeled separately and well. An issue with this might be that it isn’t robust to noise and which might affect the performance.