Departmemt of Mathematics and Statistics

CAPSTONE PROJECT

Comparison of Holt's Winter and LSTM-RNN methods for Univariate Time Series

Michael Carnival

Supervisor: **Dr. Shusen PU**

November 20, 2024

Contents

CHAPTER 1

Abstract

The explosion of data in our era presents limitless opportunities for time-series analysis. This study examines the spread of monkeypox virus on a daily basis and Florida's monthly flood numbers, areas crucial for individual livelihoods. Accurate modeling is vital for disaster preparedness and response. This study investigates the performance of Holt-Winter's method and Recurrent Neural Networks with Long Short-Term Memory (LSTM). The Holt-Winter additive and RNN-LSTM methods were applied to the monkeypox dataset and flood dataset. The data preprocessing included aggregation of the data by date and daily new cases and preceded time-series plotting to identify trends and seasonal patterns. The monkeypox dataset exhibited a 7-day seasonal component, confirmed by autocorrelation analysis. Data was split into 95% training and 5% testing sets. The Holt-Winter additive (HWA) model outperformed RNN-LSTM, yielding 55.75 RMSE versus 68.27 RMSE. Both model also struggled to account for the addition seasonal component present in the dataset. For the Flood analysis, both model posed challenges; monthly aggregated data showed no trend or seasonal patterns. Both models struggled to predict future flood events due to inherent data complexities. Test set results were 11.79 RMSE for HWA and 10.19 RMSE for RNN-LSTM. Actual versus predicted monthly flood plots revealed discrepancies by a wide margin. This study highlights strengths in modeling seasonal patterns of monkeypox daily cases and weaknesses in predicting sudden natural disasters.

CHAPTER 2

Introduction

Time series data are ubiquitous in our everyday lives, with applications ranging from weather forecasting and the stock market to patient monitoring and energy consumption. A time series is a set of observations, each recorded at a specific time. The development of time series analysis began early with stochastic processes in the 1920s and 1930s, when G.U. Yule and J. Walker applied autoregressive models. During this period, the moving average was introduced to remove periodic fluctuations in time series due to seasonality. Herman Wold later introduced the AutoRegressive Moving Average (ARMA) model for stationary series but was unable to formulate a likelihood function for maximum likelihood estimation of parameters. It wasn't until 40 years later that George E.P. Box and Gwilym Jenkins formulated this function. Since then, Box-Jenkins models have been commonly used in many forecasting techniques and seasonal adjustments [1]. Time series analysis is crucial in various fields, including engineering, science, sociology, and economics. The primary goal of time series analysis and forecasting is to draw inferences by setting up a hypothetical probability model to represent the data. This model can provide a compact description of the data, such as representing electricity consumption as a sum of specified trend, seasonal, and random terms. Other applications of time series models include separating noise from signals, predicting future values, testing hypotheses, and conducting simulation studies. In the realm of data science, time series analysis generates insights by revealing patterns, trends, and anomalies in data, making accurate predictions, and driving decision-making. Common time-series models include Moving Average (MA), Autoregressive (AR), Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Exponential Smoothing models (SES) [2]. This paper explores two time series models Holt's Winter (HW) [3] and Long Short-Term Memory Recurrent Network (LSTM-RNN) [4] using Monkeypox cases and flood dataset to assess performance differences. The HW algorithm employs simple recursions that generalize exponential smoothing recursions of SES for level forecasting [5] and Holt's method for trend forecasting with exponentially weighted moving averages [6] to generate forecasts of series containing locally linear trends and seasonal patterns. The LSTM is a memory cell replacing the hidden layer of standard RNNs. Each memory cell contains a recurrent edge with a weight of 1 to overcome vanishing and exploding gradient problems. This recurrent edge and its values are known as the cell state, comprising the forget gate, input gate, and output gate. The forget gate allows the memory cell to reset its internal state when stored information is no longer necessary. Input gates are simple sigmoid threshold units with an activation function range of [0, 1], controlling when the gate opens (1) or closes (0) to receive cell state updates. The output gate learns to decide when to update hidden unit values. Through these gates, LSTM can add or remove information from the memory cell, mitigating vanishing and exploding gradients in RNN [7].

CHAPTER 3

Methodology

3.1 Data acquisition

The datasets I will be using to experiment with for the methods chosen are from ”ourworldindata.org,” on the topic of epidemiology in monkeypox virus. The structure of monkeypox dataset contains 90,061 rows by 17 columns. The date range are from May 1, 2022 to September 16, 2024.The column list include;

location, date, iso_code, total_cases, total_deaths, new_cases, new_deaths, new_cases_smoothed, new_deaths_smoothed, new_cases_per_million, new_cases_smoothed_per_million, new_deaths_per_million, total_deaths_per_million, new_deaths_smoothed_per_million, suspected_cases_cumulative, and annotation.

For the sake of simplicity, we will consider forecasting new_cases focusing on specific location namely, "world." The preprocessing steps required in this monkeypox dataset include retaining only the date (i.e., daily, weekly, or monthly), and new_cases columns as well. The dataset will be split sequentially into a training set and a testing set and follow 95% train to 5% test. In addition, the flood 1996 to 2020 datasets obtained from https://www.ncdc.noaa.gov/stormevents/ will be used as well. The structure of the stormevents dataset has 144314 rows by 55 columns. The columns list includes;

'BEGIN_YEARMONTH', 'BEGIN_DAY', 'BEGIN_TIME', 'END_YEARMONTH', 'END_DAY', 'END_TIME', 'EPISODE_ID', 'EVENT_ID', 'STATE', 'STATE_FIPS', 'YEAR', 'MONTH_NAME', 'EVENT_TYPE', 'CZ_TYPE', 'CZ_FIPS', 'CZ_NAME', 'WFO', 'BEGIN_DATE_TIME', 'CZ_TIMEZONE', 'END_DATE_TIME', 'INJURIES_DIRECT', 'INJURIES_INDIRECT', 'DEATHS_DIRECT', 'DEATHS_INDIRECT', 'DAMAGE_PROPERTY', 'DAMAGE_CROPS', 'SOURCE', 'MAGNITUDE', 'MAGNITUDE_TYPE', 'FLOOD_CAUSE', 'CATEGORY', 'TOR_F_SCALE', 'TOR_LENGTH', 'TOR_WIDTH', 'TOR_OTHER_WFO', 'TOR_OTHER_CZ_STATE', 'TOR_OTHER_CZ_FIPS', 'TOR_OTHER_CZ_NAME', 'BEGIN_RANGE', 'BEGIN_AZIMUTH', 'BEGIN_LOCATION', 'END_RANGE', 'END_AZIMUTH', 'END_LOCATION', 'BEGIN_LAT', 'BEGIN_LON', 'END_LAT', 'END_LON', 'EPISODE_NARRATIVE', 'EVENT_NARRATIVE', 'DATA_SOURCE', 'datetime_begin', 'datetime_end', 'fatality', 'damages'

For our purpose, the EVENT_TYPE = 'flood', and the END_YEARMONTH columns are retained for analysis and forecasting. The preprocessing procedure includes aggregating the datetype into either days or months and separating the flood types into flood that consist of categories of flashflood, floods, coastal floods, and lakeshore for their own analysis. The splitting of the dataset for each category will have the same train test split of 95% and 5%, respectively.

3.2 Holt's Winter

The Holt's Winter (HW) method is a double exponential smoothing technique for trended and seasonal time series. It has been widely used tool for forecasting in business data dealing with changing trends and seasonal correlation. The HW have many variations and the component of each are different. For the first variation, additive HW method, the seasonal fluctuations are constant and follows the equations below

$$L_t = \alpha(Y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t_1} \tag{3.1}$$

$$b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1} \tag{3.2}$$

$$S_t = \gamma(L_t - L_t) + (1-\gamma)S_{t-s} \tag{3.3}$$

$$F_{t+m} = L_t + b_t m + S_{t-s+m} \tag{3.4}$$

Where L_t is level, b_t is trend, S_t is the seasonality, and F_{t+m} is forecast. α (associated with level), β (associated with trend) and γ are the smoothing parameters, m is the number of forecasts ahead, s is the season's length (e.g., monthly, weekly, daily), and Y_t is the observed data at time point t [3]. For the second variation, the multiplicative method, the seasonal fluctuations change proportionally to the level of the series and follows equations below.

$$L_t = \alpha(Y_t/S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t_1} \tag{3.5}$$

$$b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1} \tag{3.6}$$

$$S_t = \gamma(L_t/L_t) + (1-\gamma)S_{t-s} \tag{3.7}$$

$$F_{t+m} = (L_t + b_t m)S_{t-s+m} \tag{3.8}$$

3.3 Long-Short Term Memory

The Long short-term memory (LSTM) framework is another popular model for time-series forecasting. It is a type of Recurrent Neural Network (RNN) that leverages previous output states as inputs to update the current hidden state. This process unfolds sequentially, with each time step building upon the previous one (a_t becomes a_{t-1}). The network iterates through all time steps, refining its hidden state until it reaches the final state. This final state is then used to generate an output. The output is compared to the actual value, and the error is computed. Through backpropagation, the network updates its weights and biases to minimize this error. Notably, the RNN's weights and biases remain shared across all inputs and time steps, facilitating efficient learning. The traditional RNN architecture has a significant limitation: as we unfold more RNN layers, the infamous problem, vanishing and exploding gradient problems emerges [4]. Specifically, the vanishing gradient problem occurs when gradients become increasingly small, leading to infinitesimally small update steps that prevent the model from reaching its optimal parameters. Conversely, the exploding gradient problem arises when gradients grow excessively large, causing the model to take oversized steps and bounce erratically around the optimization curve, never converging on the optimal parameters. Luckily, the LSTM addresses these issues. It puts forward memory cells (cell state) that is able to capture and store temporal patterns in sequential data. The way LSTM manage information is by having three control gates. The first gate is the Forget Gate, which controls what information is forgotten from the memory cell state. The ins and outs of a forget gate work by computing the weighted sum of input features and previous hidden state. Then it applies a sigmoid activation function to get a value between 0 (get rid of this), and 1 (keep this). Lastly, it is multiplied by the previous memory cell state. The equation of this gate follows.

$$f_t = \sigma(W_f \dot{[} h_{t-1}, x_t] + b_f) \tag{3.9}$$

Where t is time, f_t denotes the forget gate, σ is the sigmoid activation function, W_f is the a set of weight, h_{t-1} is the hidden units at time $t-1$ and b_f is the bias vector.The second gate is the input gate, which decides what new information will be stored in the cell state. This gate has two parts, a sigmoid layer (input gate layer) that decides what values gets updated, and a tanh layer that produces a vector of new applicant values (1 and -1), $\tilde{c}_t$ for the new cell state (i.e., adding new applicants to the cell state replacing the old one we forgot). The equation of this gate follows.

$$I_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \tag{3.10}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3.11}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.12}$$

Where i_t is the input gate, $\tilde{C}_t$ is the candidate value, C_t is the new cell state, tanh is the activation function (of a value between (-1 and 1), W_c and b_C are the weights and bias for the new cells state. The third gate is the output gate, which decides the output based on the cell state in a filtered version. It works by passing through a sigmoid layer which controls what parts of the cell state gets outputted and passed through tanh activation function then is multiplied by the sigmoid gate. The equations are as follows.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.13}$$

$$h_t = o_t * tanh(C_t) \tag{3.14}$$

Where o_t is the output gate, W_o and b_o are the weights matrix for the output gate and the bias vector for output gate, and h_t is the hidden units at current time step.

3.4 Model Evaluation

Through these two methodologies, we will investigate the performance of Holt's winter and LSTM methods on both the monkeypox and the flood datasets for their accuracy. To evaluate the performance of our methods, the Root-Mean-Square Error (RMSE) will be used. The RMSE is a popular metric for judging the accuracy of a model by measuring the residuals between the actual and predicted values. The RMSE formula is computed as follows

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \tag{3.15}$$

Where N is the length of the observation, x_i is the target value, $\hat{x}_i$ is the estimated value.

3.5 Programming of the Method

The software tools used to apply these methods is through python platform with libraries; numpy, pandas, scikitlearn, tensorflow, and statsmodel. This project was done locally. The potential for ethnical concerns include privacy issue related to the people who have contacted monkeypox disease. Luckily, no obvious identifiers are discernible to anyone in the dataset.

CHAPTER 4

Results

4.1 Holt's Winter Additive Model

4.1.1 Monkeypox New Cases analysis

We evaluated the HWA model using the monkeypox dataset, focusing on the "new_cases" variable for the "world" group. The timeline spanned from 2023-10-09 to 2024-07-21. The resulting time-series plot is shown in Figure 4.1.

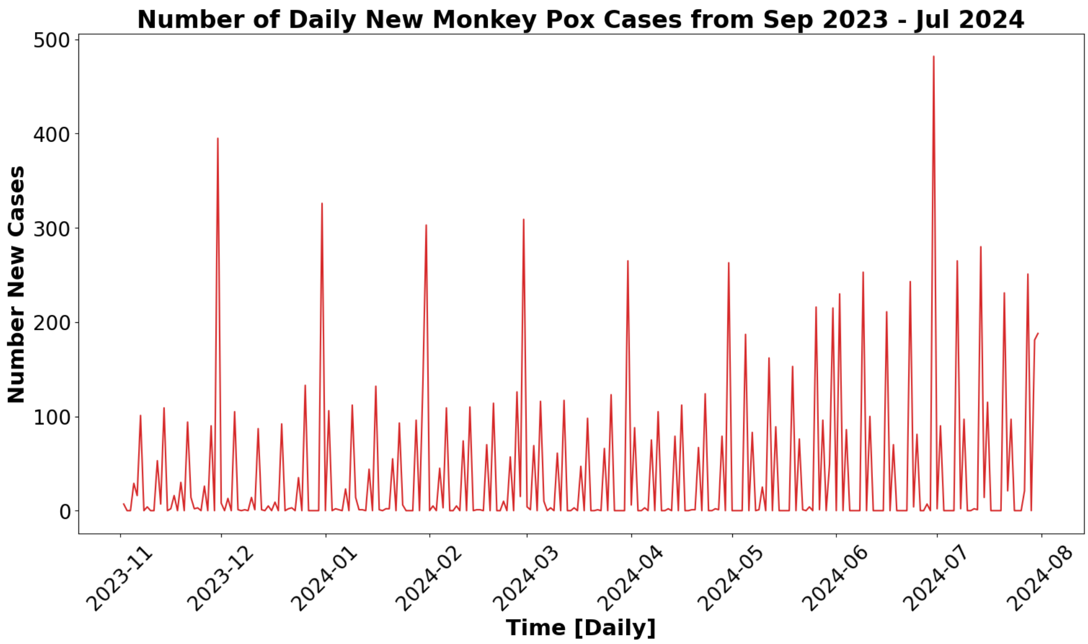

Figure 4.1: 2023-10 to 2024-8 Daily Mpox Time-series

This time-series plot exhibits a clear seasonal pattern with peaks occurring every 7 days and no obvious trend. To confirm this periodicity, we plotted the autocorrelation function Figure 4.2.

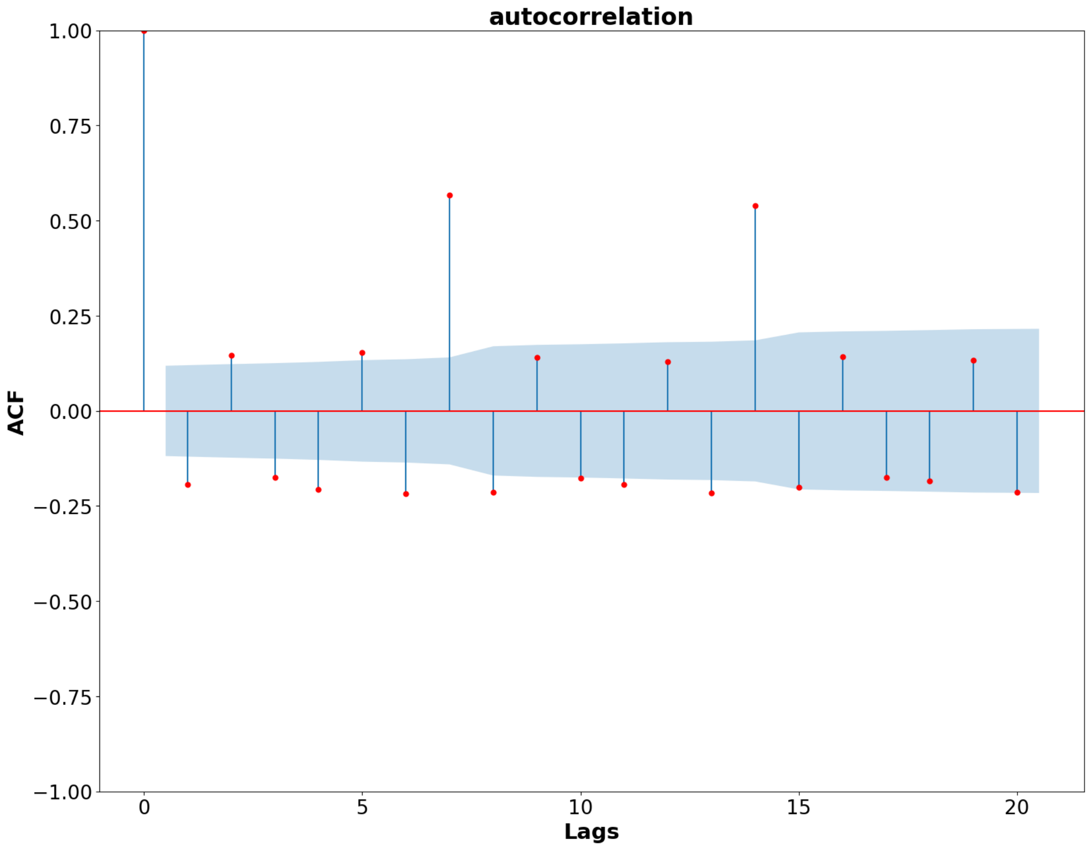

Figure 4.2: Autocorrelation Plot of the Mpox Time-series

Typically, modeling time-series data requires stationarity. Fortunately, the HWA model does not require this criterion. Following the method section guidelines, we split the data into 95% training and 5% testing sets, yielding 272 training observations and 13 testing observations. Utilizing the Statsmodels.api library and exponential smoothing module, we set seasonal_periods 7̄ and seasonal "additive." Model parameters was fit optimally by setting fit(optimized =True) to obtain smoothing levels of 0.004, seasonal smoothing 0.194 and an initial level of 30.98. The Predicted and actual values are compared in Figure 4.3.

The actual test data is represented by the blue series, while predicted test data is indicated by the orange series. The HWA model achieved an RMSE of 55.75 for the test set.

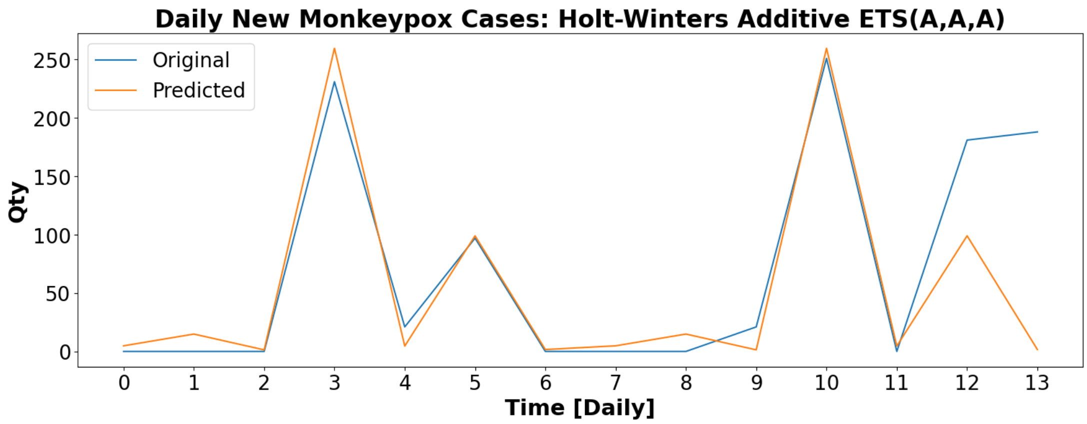

Figure 4.3: HWA: Daily Actual Test Data vs the Predicted New Mpox cases.

4.1.2 Flood Count Analysis

For the flood count dataset, we selected monthly data from 1996-01 to 2020-07. The train/test split was 95% and 5%, respectively. As shown in Figure 4.4.,

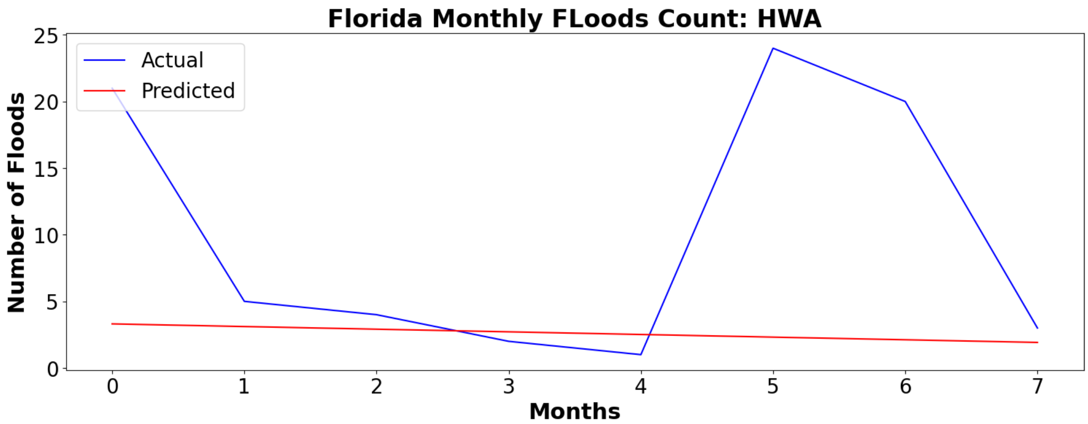

Figure 4.4: HWA: Monthly Actual Test Data vs the Predicted Floods Numbers

the HWA model poorly fits the flood count actual test set, particularly at the beginning and months 4 through 7. The RMSE for this model was 11.79.

4.2 RNN-LSTM Model

4.2.1 Monkeypox New Cases analysis

Using the same timeline and train/test split as the HWA model, the RNN-LSTM method's parameters were set to timestep = 7, five LSTM layers (256, 128, 64, 32, 16 units), dropout = 0.2, a dense layer with 1 output, ADAM optimizer, MSE loss function, RMSE metric and 200 epochs. Actual test and predicted values are plotted in Figure 4.5.

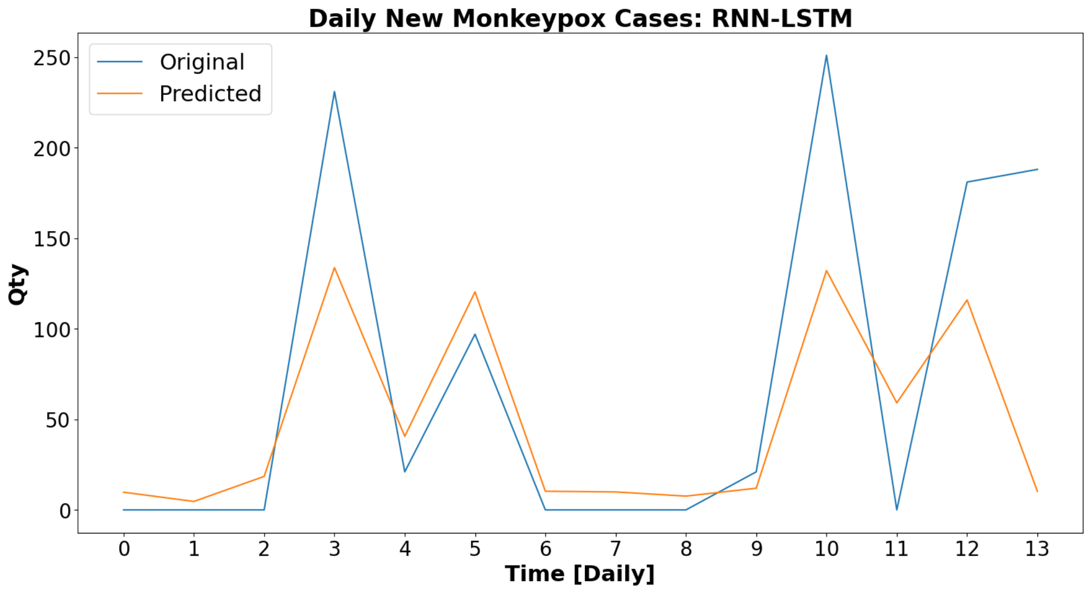

Figure 4.5: Daily Actual Test Data vs the Predicted New Mpox cases.

The RMSE resulted in 68.27 for the test set.

4.2.2 Flood Count Analysis

On the flood count dataset with identical time range and 95%/5% train/test split, the RNN-LSTM performed poorly shown in Figure 4.6, achieving an RMSE of 10.19, slightly better than the HWA model.

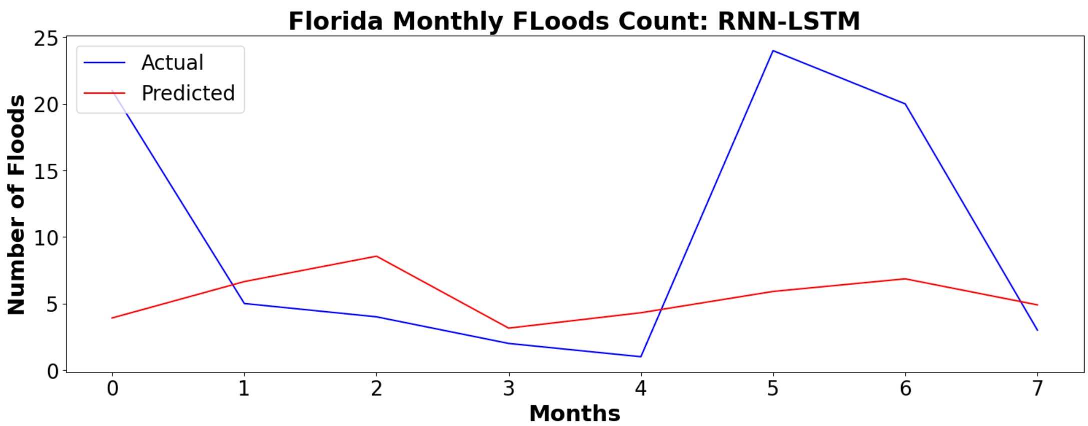

Figure 4.6: RNN-LSTM: Actual Test Data vs the Predicted Number of Floods.

4.3 Comparison of Models on the Training data

The Figure 4.7 shows training performance between models on the monkeypox dataset. The RNN-LSTM's fitted training set values are consistent but fail to capture increasing new case counts starting from day 200. In contrast, the HWA method's fitted values increased from day 200, better representing the trend. Training RMSE values were 42.36 for RNN-LSTM and 53.23 for HWA.

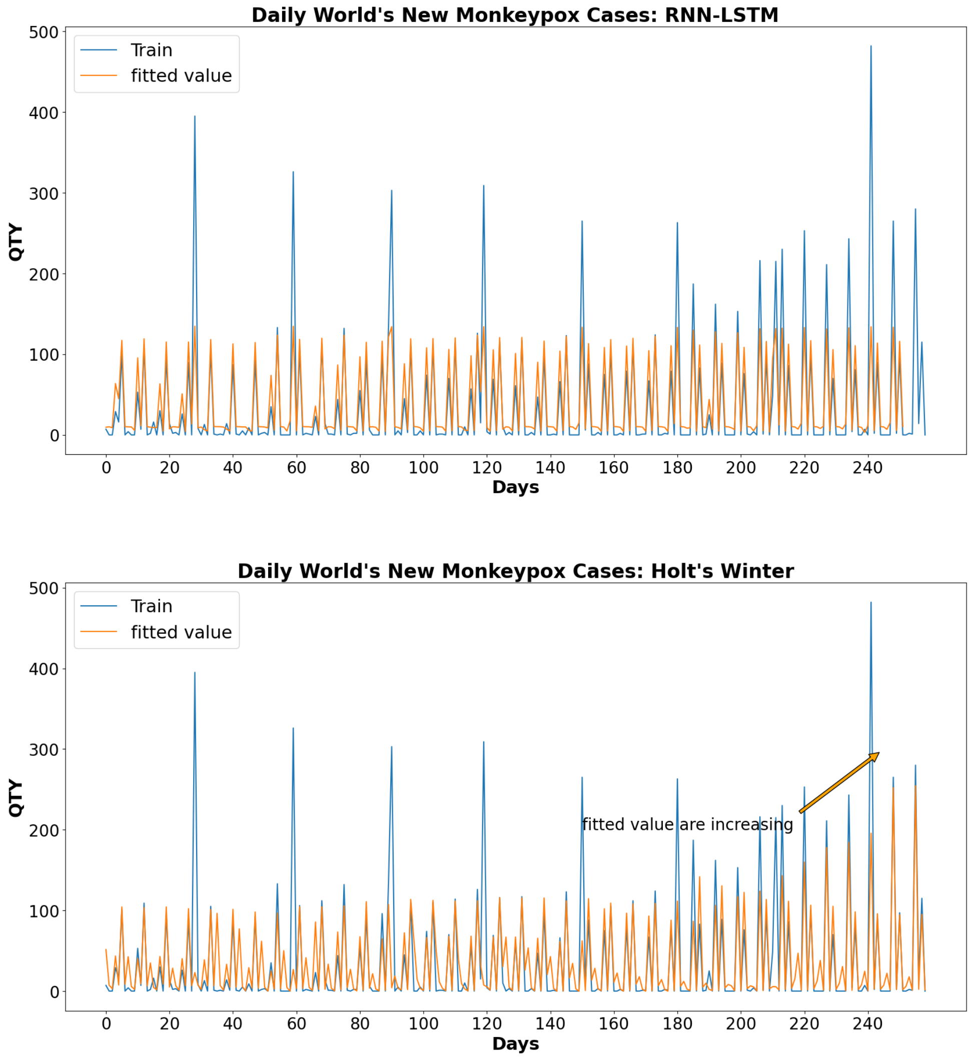

Figure 4.7: MPox Newcase Count Comparsion of Actual Train Data vs. Fitted Data.

The RNN-LSTM's inferior test set performance and lower training RMSE suggest potential overfitting compared to the HWA model. Both models struggle to capture large spikes in monkeypox new cases. These uncaptured spike suggest an additional seasonality besides the weekly pattern.

For the flood dataset, the Figure 4.8 does not clearly indicate which model performs better in training data of monthly flood count.

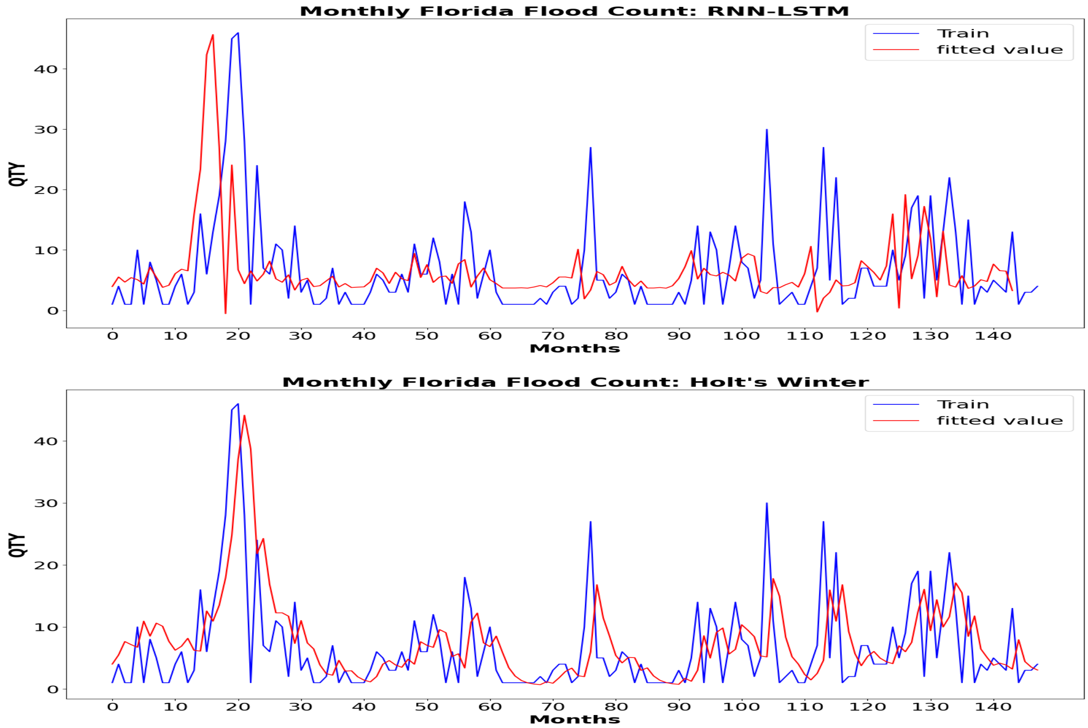

Figure 4.8: Monthly Flood Comparsion of Actual Train Data vs. Fitted Data.

The Calculated RMSE values for training data predictions were 9.02 for RNN-LSTM and 7.66 for HWA. Although, the HWA has less rmse compared to the RNN-LSTM, both models face limitations in capturing the unpredictable nature of future flood occurrences.

CHAPTER 5

Discussion and Future Work

In our monkeypox analysis, we found that forecasting future cases was significantly easier using the HWA and RNN-LSTM method. However, both methods struggled to capture additional seasonal patterns in the series. The HWA outperformed the RNN-LSTM on the test set by following the increasing frequency of new cases more accurately.

This study demonstrates Holt-Winters Additive method's exceptional performance in forecasting data exhibiting seasonality and trends, without prior data assumptions. These findings underscore HWA's predictive prowess for similar datasets. Beyond epidemiology, HWA's strengths benefit business decision-making, where seasonal fluctuations and trends are ubiquitous.

For future forecasting monkeypox case, other algorithm like the NeuralProphet presents a promising algorithm, and SARIMA analysis warrants consideration. Also, incorporating multiple factors can enhance prediction accuracy [8].

Conversely, both methods failed to accurately fit the Florida monthly flood count test set. Although the RNN-LSTM yielded slightly better RMSE than the HWA, it was insufficient to be compete against hydrodynamic models. Forecasting future flood counts is inherently challenging due to dependence on diverse variables (e.g., drainage basins, data scarcity and inadequate drainage networks) [9]. Current state-of-the-art hydrodynamic model frameworks surpass most machine learning (ML) approaches.

Nevertheless, ongoing research into neural network architectures, fueled by recent ML advancements, offers hope for more robust predictions [10].

Although RNN-LSTM didn't surpass Holt-Winters Additive for Mpox data, its relevance spans diverse fields. Notably, augmented LSTM models are being integrated into software to solve differential equations modeling natural processes in life sciences, physics and neural networks. Ongoing innovations underscore RNN-LSTM's potential[11].

This study reveals that monkeypox virus transmission exhibits predictability using current forecasting techniques, whereas predicting monthly flood counts remains exceedingly difficult with existing methods.

APPENDIX A

Appendix Table

Table A.1: RMSE Performance Comparison

Method	Daily Mpox Cases		Monthly FL Floods	
	Train	Test	Train	Test
HWA	53.23	55.75	7.66	11.79
RNN-LSTM	42.36	68.27	9.02	10.19

Bibliography

[1] Stockholm University. A brief history of time series analysis. *Retrieved October 1, 2024, from https://www.su.se/english/research/research-subjects/statistics/statistical-models-in-the-social-sciences/a-brief-history-of-time-series-analysis-1.612367*

[2] P.J. Brockwell and R.A. Davis. Introduction to time series and forecasting. *Springer New York*, 2016.

[3] P.R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science, 6(3)*, 324–342, 1960.

[4] S. Hochreiter. Long Short-term Memory. *Neural Computation, MIT-Press*, 1997.

[5] R.G. Brown. Statistical forecasting for inventory control. *McGraw-Hill*, 443–473, 1959.

[6] C.C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting, 20(1)*, 5–10, 2004.

[7] R.C. Staudemeyer and E.R. Morris. Understanding LSTM–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

[8] D. Long, B. Bowen, F. Tan and M. Newman. Forecasting the monkeypox outbreak using ARIMA, prophet, NeuralProphet, and LSTM models in the United States. *Forecasting* 5.1, 2023, 127–137.

[9] Grey Nearing, et al. Global prediction of extreme floods in ungauged watersheds. *Nature* 627.8004, 2024, 559–563.

[10] P.B. Bedient, A. Holder, J.A. Benavides and B.E. Vieux. Radar-based flood warning system applied to Tropical Storm Allison. *Journal of Hydrologic Engineering* 8.6, 308–318.

[11] Shiri, Farhad Mortezapour, et al. A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *arXiv preprint arXiv:2305.17473*, 2023.