Daily Rainfall Time Series Analysis of Central Park, New York

Omid Emamjomehzadeh[*]
Department of Civil and Urban Engineering,
New York University

Ruixuan Zhang
Department of Civil and Urban Engineering,
New York University

Ahmadreza Ahmadjou
Department of Electrical and Computer Engineering
New Tork University

Abstract

Precipitation, a crucial component of the (urban) water cycle, holds significant implications for natural water resource management and flood-control infrastructure resilience and management. This study investigates precipitation properties and evaluates the forecasting efficacy of classical and novel methodologies. Frequency domain analysis is employed to analyze rainfall periodicity, followed by predictive modeling using ARIMAX and Marked Point Process models to forecast next-day precipitation. The Long Short-Term Memory (LSTM) network is also implemented with hyper-parameter tuning to capture the temporal correlations between different meteorological variables. Uncertainty in LSTM predictions is quantified using the Gaussian Process model of the residuals. The methods employed to dataset four meteorological variables measured in Central Park, New York City. Frequency analysis reveals the lack of periodicity in precipitation and highlights the challenges posed by impulses. For forecasting, the ARIMAX model could capture the trend. The Mean Squared Error (MSE) for the forecasting is 4.01, and the Mean Absolute Percentage Error (MAPE) is 42.8%. While the SARIMAX model is built upon normal distributions, The Marked Point Process provided intuition about the impulse-like behavior of the measurement data. Finally, the study also shows low accuracy for LSTM to model exact values, but it can model the trends; however, the Gaussian process can model the residual accurately.

Introduction

Forecasting meteorological variables, especially precipitation, is a vital field of study with significant implications for urban, environmental, and economic planning. Precipitation, an essential element of the hydrological cycle, directly impacts urban flood control infrastructures, agriculture, and water resources. One of the implications of accurate precipitation forecasting is that early warning systems can use that information to reduce the impacts of flooding by delivering prompt information for emergency interventions and preventative actions (1). Accordingly, comprehending rainfall stochastic characteristics is essential for developing a resilient urban drainage system capable of handling standard and extreme precipitation events.

The initial approach entails developing physics-based models based on differential equations, which are numerically solved to replicate atmospheric dynamics. This methodology has been fundamental to meteorological forecasting for decades, with countries creating advanced global circulation

[*]omid.emamjomehzadeh@nyu.edu

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

models (GCMs) for several applications, exemplified by NOAA's Geophysical Fluid Dynamics Laboratory model in the United States. The second way utilizes machine learning and probabilistic techniques to capture stochastic rainfall characteristics. The latest approach incorporates physics into machine learning models, as demonstrated by breakthroughs such as *Neural GCM* and *GraphCast*, which exemplify advanced developments in weather prediction by merging classical physics with modern data-driven techniques (2; 3).

The main aim of this study is to examine precipitation properties and evaluate the forecasting efficacy of some probabilistic and deep-learning models. The study aims to analyze precipitation periodicity using Fourier analysis. Subsequently, predictive modeling is performed with the SARIMAX model to predict future precipitation, and further predictive analysis is done using the Marked Point Process. The research subsequently assesses the efficacy of a deep learning methodology by implementing hyper-parameter tuning and doing uncertainty analysis of a Long Short-Term Memory (LSTM) network. The project ultimately evaluates the precision and dependability of various probabilistic and deep learning methodologies.

This document is structured as follows. The Methodology section provides a detailed explanation of the strategies and techniques employed to analyze rainfall characteristics and implement the forecasting models. The Dataset section delineates the data utilized in this research, emphasizing its principal characteristics. The Results section presents the findings for each method described in the methodology, encompassing performance metrics and uncertainty analysis. The Discussion and Conclusion section ultimately evaluates the meaning of the findings, recognizes limits, and suggests avenues for future research.

Methods and Algorithms

Frequency domain analysis

As discussed in the lecture and related time series forecasting literature, frequency domain analysis is useful for identifying whether the subject time series is dominated by periodic patterns. If such patterns are identified from the raw noise inputs, they can significantly simplify the forecasting task. Additionally, frequency domain analysis offers an alternative perspective on the time series, providing an intuitive understanding of its complexity and why certain methods may fail. Specifically, we apply the *Discrete Fourier Transform (DFT)*, as shown in Eq. (1), and the *Discrete Short-time Fourier Transform (DSTFT)*, as shown in Eq. (2)-(3), to the target time series.

$$\tilde{X}_n = \sum_{t=0}^{T-1} X_t e^{\frac{-2\pi i n t}{T}} \tag{1}$$

$$\tilde{X}(k, t_w) = \sum_{t=0}^{T-1} w_{t_w}(t; l) X_t e^{\frac{-2\pi i n t}{T}} \tag{2}$$

$$w_{t_w}(t; l) = \begin{cases} 1 & |t - t_w| \leq l \\ 0 & |t - t_w| > l \end{cases} \tag{3}$$

The temporal resolution also affects frequency domain analysis results. Several temporal aggregations are applied, as shown in Eq. (4), where H is the aggregation window and M is the total number of data points after aggregation. As the first step in understanding the intrinsic characteristics of time series, frequency domain analysis lays the foundation for selecting appropriate time-series forecasting methods and interpreting possible failures.

$$\hat{X}_m = \frac{1}{H} \sum_{j=1}^{H} X_{H(m-1)+j}, \quad m = 1, 2, 3, \cdots, M \tag{4}$$

SARIMAX models

Based on the nature of the measurement data in weather forecasting, there are three factors that need to be considered: 1) Seasonality of weather measurements, 2) the contribution of different elements

like temperature on precipitation, and 3) whether the time series is stationary or not. Accordingly. Seasonal Autoregressive Integrated Moving Average with eXogenous variables (SARIMAX) models can be used to address each of the above criteria. This model builds upon the widely used ARMA models. The following provides an overview and development of this model (4):

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}. \tag{5}$$

In operator form:

$$\Phi(\mathcal{B}) y_t = c + \Theta(\mathcal{B}) \varepsilon_t, \tag{6}$$

where $\Phi(\mathcal{B}) = 1 - \phi_1 \mathcal{B} - \cdots - \phi_p \mathcal{B}^p$ and $\Theta(\mathcal{B}) = 1 + \theta_1 \mathcal{B} + \cdots + \theta_q \mathcal{B}^q$, and $\epsilon \sim N(0, \sigma_w^2)$. ARMA models are used mainly to model stationary series that exhibit a clear autocorrelation structure. However, many real-world time series, such as weather measurements, may not follow the stationary criteria required for the ARMA model. Therefore, differencing the series can remove these non-stationary components. The differencing operator is defined as:

$$\nabla y_t = y_t - y_{t-1}, \quad \text{and more generally } \nabla^d y_t = (1 - \mathcal{B})^d y_t. \tag{7}$$

In accordance with the differencing operator, the ARIMA(p, d, q) model applies the ARMA(p, q) framework to the differenced series $\nabla^d y_t$, yielding:

$$\Phi(\mathcal{B})(1 - \mathcal{B})^d y_t = c + \Theta(\mathcal{B}) \varepsilon_t. \tag{8}$$

By appropriately choosing d, ARIMA models convert non-stationary processes into stationary ones, enabling the ARMA machinery to operate effectively on the transformed data. Precipitation exhibits seasonal patterns that repeat every s period where $s = 12$. To handle seasonality, we introduce seasonal differencing and seasonal ARMA terms:

$$(1 - \mathcal{B}^s)^D \tag{9}$$

is the seasonal differencing operator of order D, which removes seasonal patterns from the data. Seasonal AR and MA polynomials are defined in a similar manner but with terms involving $\mathcal{B}^s$ instead of $\mathcal{B}$. A seasonal ARIMA model, denoted as ARIMA$(p, d, q)(P, D, Q)_s$, generalizes ARIMA to include seasonal effects:

$$\Phi(\mathcal{B}^s)\Phi(\mathcal{B})(1 - \mathcal{B})^d (1 - \mathcal{B}^s)^D y_t = c + \Theta(\mathcal{B}^s)\Theta(\mathcal{B})\varepsilon_t, \tag{10}$$

where:

- $\Phi(\mathcal{B}^s)$ and $\Theta(\mathcal{B}^s)$ are the seasonal AR and MA polynomials.
- $(1 - \mathcal{B}^s)^D$ applies seasonal differencing to remove periodic patterns.

By capturing both the non-seasonal and seasonal components, SARIMA models are adept at handling complex, regularly repeating patterns found in many practical time series. While SARIMA models rely purely on the internal structure of the series, many time series are influenced by external factors. This factor could be weather-related variables such as snow depth and temperature. Incorporating these external influences can significantly improve forecasting accuracy and interpretation. Let $\mathbf{x}_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,k})$ represent a vector of exogenous regressors at time t, and let $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)$ be their associated coefficients. By introducing these regressors into the SARIMA framework, we obtain SARIMAX:

$$\Phi(\mathcal{B}^s)\Phi(\mathcal{B})(1 - \mathcal{B})^d (1 - \mathcal{B}^s)^D (y_t - \mathbf{x}_t^\top \boldsymbol{\beta}) = c + \Theta(\mathcal{B}^s)\Theta(\mathcal{B})\varepsilon_t. \tag{11}$$

Here, $y_t - \mathbf{x}_t^\top \boldsymbol{\beta}$ can be viewed as the response variable after adjusting for known external effects. The remaining dynamics, including seasonal and non-seasonal ARMA-like structures, model the residual variability once these external factors have been accounted for. The parameters in SARIMAX models are commonly estimated via maximum likelihood methods, often implemented using numerical optimization techniques. In this study, the *statsmodels* package in Python was utilized to implement the desired SARIMAx model.

Marked Point Process Decoder

The occurrence of precipitation can be viewed as distanced impulses distributed over time. Considering the daily rain and snowfall, one can expect a sparse impulse train for participation over the year. The pattern includes a concentration of impulses over some interval, specifically in the fall and winter seasons, followed by sparse events during spring and summer. In this regard, a marked point process (MPP)-based Bayesian filtering approach is introduced in (5). In this approach, the time axis is divided into small, evenly-spaced time bins with the duration T_s. The occurrence of an impulse in the i^{th} bin is indicated via a binary value, with '1' indicating the existence of an impulse and '0' indicating the absence of the impulse. The probability of an impulse happening is a random variable with Bernoulli distribution, with probability mass function $p_i^{n_i}(1-p_i)^{1-n_i}$.
The state-space model incorporates a state equation that characterizes arousal dynamics through a stochastic random walk process, i.e,

$$a_i = a_{i-1} + \xi_i \tag{12}$$

where a_i is the hidden state and $\xi_i \sim \mathcal{N}(0, \sigma_\xi^2)$ is the process noise. Consequently, a sigmoid function is used to establish a relationship between p_i and a_i,

$$\log \frac{p_i}{1-p_i} = \beta + a_i \Rightarrow p_i = \frac{1}{1+e^{-(\beta+a_i)}} \tag{13}$$

The employment of the sigmoid function guarantees the probabilistic nature of p_i. In this equation, the probability of an impulse occurring increases with higher levels of a continuous hidden state, and conversely, it decreases when the state level falls. Here, β is an unknown constant to be estimated. The constant β is estimated by considering the initialization of a_i near 0. Hence,

$$\beta \approx \log(\frac{p_0}{1-p_0}) \tag{14}$$

The value of p_0 can be approximated by calculating the average probability of an impulse occurring throughout the dataset. Now, an amplitude q_i is considered for each impulse. A linear relationship is defined between the arousal state and the amplitudes,

$$q_i = \gamma_0 + \gamma_1 a_i + \nu_i \tag{15}$$

where, γ_0 and γ_1 are the values to be estimated. Therefore, one can express a joint density function for the as follows,

$$p(n_i \cap q_i \mid a_i) = \begin{cases} 1-p_i & n_i = 0 \\ p_i \frac{1}{\sqrt{2\pi\sigma_\nu^2}} e^{\frac{-(q_i-\gamma_0-\gamma_1 a_i)^2}{2\sigma_\nu^2}} & n_i = 1 \end{cases} \tag{16}$$

The hidden state state a_i and parameters of the model γ_0, γ_1, σ_ν^2 and σ_ξ^2 are to be determined. According to Eq (12) - (13), and (15) - (16), the unknown parameters are estimated by maximizing the following expected log-likelihood,

$$\begin{aligned} J = &\sum_{i=1}^{I} \mathbb{E}[n_i(\beta + a_i) - \log(1 + e^{\beta + x_i})] \\ &- \frac{I}{2}\log(2\sigma_\nu^2) - \sum_{i=1}^{I} \frac{\mathbb{E}[(q_i - \gamma_0 - \gamma_1 a_i)^2]}{2\sigma_\nu^2} \\ &- \frac{I}{2}\log(2\sigma_\xi^2) - \sum_{i=1}^{I} \frac{\mathbb{E}[(a_i - a_{i-1})^2]}{2\sigma_\xi^2} \end{aligned} \tag{17}$$

For the MLE problem (17), a Bayesian filtering approach is employed to estimate the latent state a_i within the EM scheme. The resulting EM equations to maximize this log-likelihood are detailed in (5); for brevity, the specifics are not reiterated here.

LSTM predictions and Gaussian process model of residuals

To better capture complicated precipitation patterns, time-based features are generated that encompass day of year, month, year, and season. The missing values of meteorological variables are addressed

by using a combination of temporal interpolation methods based on near values of the same variables (for example, backward-filling, and forward-filling methodologies). Additionally, seven-day rolling averages are calculated and added to the dataset since these features identify short-term trends.

The features are normalized by a standardization technique. Then the fixed-length input-output sequences are generated, with each input sequence comprising observations over a designated interval (e.g., 30 days), and the corresponding output being the observation directly after the interval. Afterward, the dataset is divided into training and validation subsets, designating 80% of the sequences for training and 20% for validation. A custom data set class was developed to enable efficient loading of these sequences into *PyTorch DataLoader* instances. The class converts input sequences and targets into tensors, facilitating efficient batching and shuffling during training.

Then, a Long Short-Term Memory (LSTM) network is developed to capture temporal dependencies in meteorological data. The architecture includes a bidirectional LSTM layer, succeeding by a fully connected layer for output prediction. Dropout regularization is implemented to alleviate over-fitting. The model utilizes the Adam optimizer to minimize the Mean Squared Error (MSE) loss. The training entails iterating over mini-batches, calculating gradients, and adjusting model parameters. Gradient clipping is utilized to avert bursting gradients. A learning rate scheduler dynamically modifies the learning rate according to validation performance.

An early-stop mechanism tracks validation loss to avert over-fitting. The training is terminated if the validation loss does not improve after a specified patience level. The validation performance is assessed after each epoch and the model with the minimum validation loss is preserved. The training progress, encompassing epoch-specific training and validation loss, is recorded to evaluate convergence. The hyper-parameters of the LSTM model namely (the number of layers, hidden size, dropout rate, learning rate, and batch size) are optimized using *Optuna* to perform efficient hyper-parameter optimization by defining an objective function which is the minimization of validation MSE.

The Residuals between observed and predicted values (generated by the LSTM) are modeled utilizing a Gaussian Process (GP) executed with *GPyTorch*. The data preparation process entails computing residuals and transforming them into PyTorch tensors. Inputs denote time indices, and outputs represent the residuals. A Gaussian Process model is developed with a zero-mean prior and a covariance function that integrates a ScaleKernel and an RBFKernel. The model is trained by maximizing the marginal log-likelihood across 100 iterations utilizing the Adam optimizer. Loss values are observed throughout training to guarantee convergence. The mean of GP predictions offered point estimates, whereas the standard deviation measures the uncertainty of the LSTM model.

Dataset

We employ the Central Park daily weather dataset, encompassing more than 140 years of daily weather observations from 1870 to 2022, sourced from a weather station in Central Park, New York (40.785091°N, -73.968285°E). This dataset comprises essential meteorological variables, including daily precipitation (PRCP), snowfall (SNOW), snow depth (SNWD), minimum temperature (TMIN), and maximum temperature (TMAX). The dataset is available to the public on Kaggle.

Fig. 1 shows a time series plot of the variables and Fig. 2 shows pair-wise scatter plots of the variables and the shape of marginal probability distribution functions of these variables. This dataset is quite new, so limited studies have been performed with it, presenting a chance to extract fresh insights about climatic and meteorological patterns in the York City region.

Experiments and results

This section presents the results of applying each model from the methodology to the dataset described in the dataset section.

Fourier domain analysis

To explore the underlying periodicity in the precipitation time series, we examine three temporal resolutions: daily, monthly, and annually. The Discrete Fourier Transform is then applied to these three-time series, as shown in Fig. 3. From the first column of the plot, we observe that daily

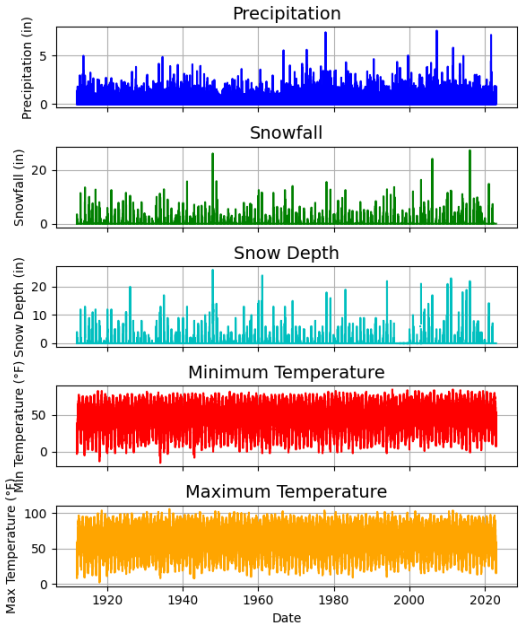

Figure 1: Time-Series plot of daily precipitation, daily snowfall, daily snow depth, minimum and maximum daily temperature in Central Park (1912-2022)

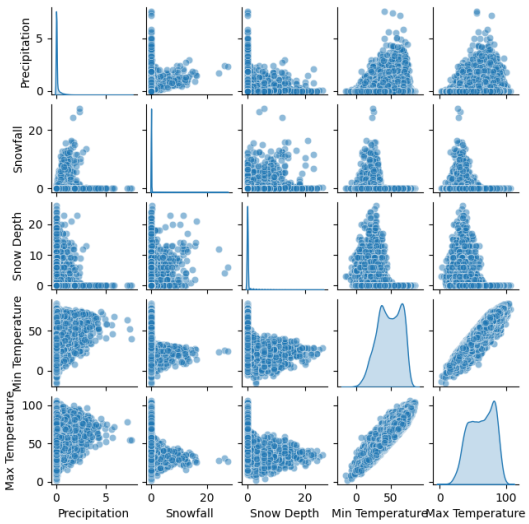

Figure 2: Scatter plots and marginal PDFs of meteorological variables (1912–2022)

precipitation consists of many "impulses" and "zeros." When temporal aggregation is applied to monthly and annual resolutions, the "zeros" are mitigated, but "impulses" remain. Visualizations in the frequency domain support this observation: regardless of the temporal resolution, amplitudes at most frequencies, except zero, are small and clustered around zero. This suggests that the precipitation data exhibit a more prominent trend rather than periodic patterns. To better understand the non-periodic nature of the data, we plot the reconstruction error using different numbers of frequencies compared to the original time series for all three temporal resolutions, as shown in Fig. 4. The reconstruction errors approach zero without a sharp drop, further confirming that the precipitation data are not dominated by a few periodic components.

We further apply the Discrete Short-time Fourier Transform to investigate any temporal-dependent periodic patterns. Three temporal resolutions are examined, and the results are shown in Fig. 5.

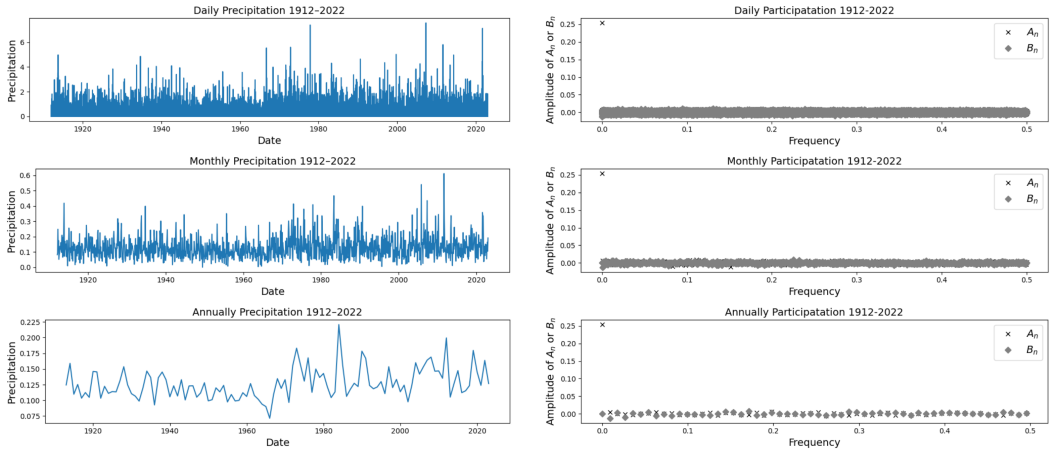

Figure 3: Visualization of time series in temporal and frequency domains.

Similar to the observations from the Discrete Fourier Transform results, most non-zero magnitudes occur at frequency zero, corresponding to the DC component. This indicates that the precipitation data are primarily dominated by offsets rather than periodic components, particularly as the data are aggregated into coarser temporal resolutions. Interestingly, in the plot using annually aggregated data, the magnitudes intensified in recent decades, which may provide an implicit clue regarding climate change and global warming.

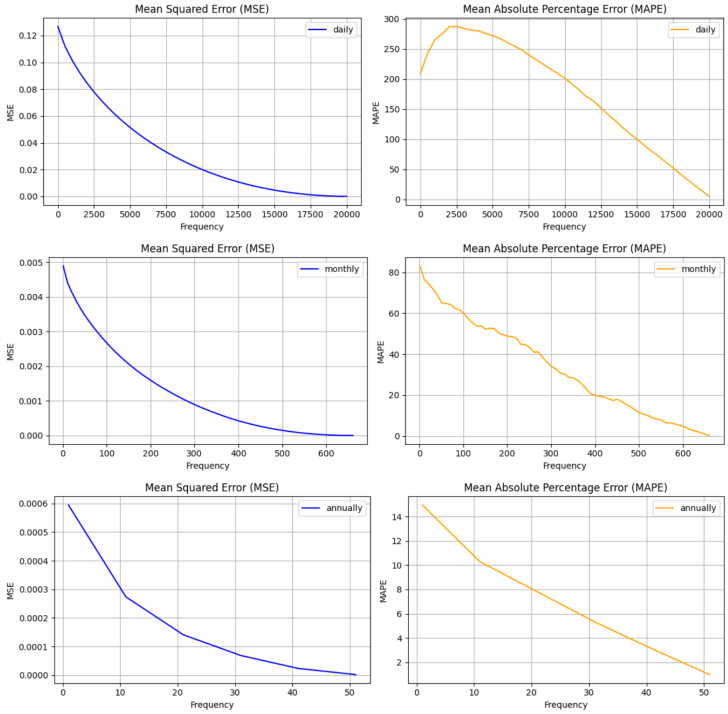

Figure 4: Reconstruction errors in MSE and MAPE under three temporal resolutions: daily, monthly, and annually. The bump in the daily MAPE plot is because of the elimination of zero values.

The frequency domain analysis provides insights into what to expect when performing forecasting tasks on this specific time series data. The key takeaways are: (1) precipitation quantity primarily exhibits offsets without significant periodic signals; and (2) the impulses in the time series pose a

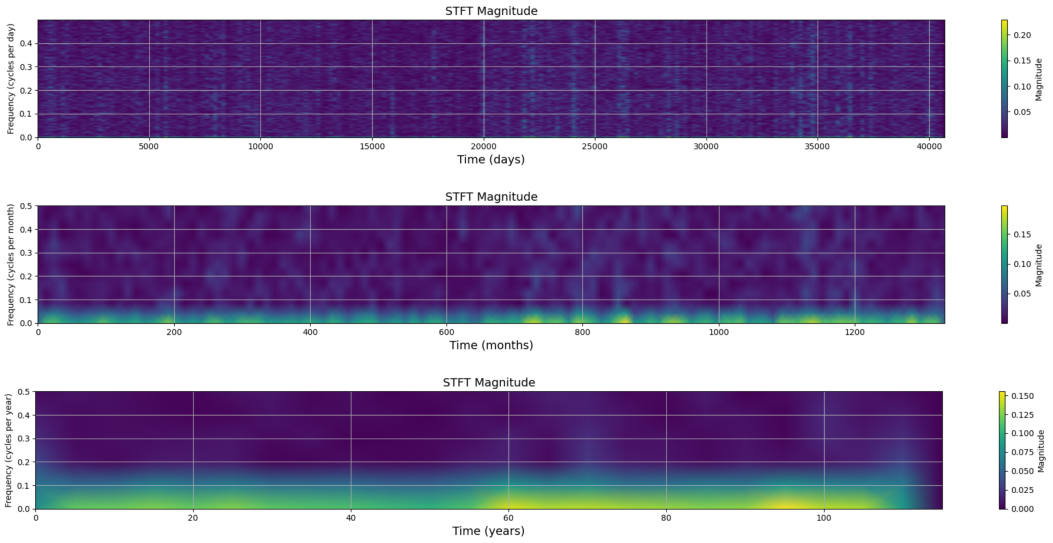

Figure 5: Visualization of magnitudes and frequencies at each time segment after applying Discrete Short-time Fourier Transform on three temporal resolutions (from top to bottom): daily, monthly, and annually.

significant challenge to many forecasting models, which tend to perform better with stationary data. Using a lower temporal resolution may help mitigate this issue.

Weather Forecasting Using SARIMAX

Before applying the SARIMAX method introduced in the previous section to the dataset, the daily data is aggregated into monthly data by aggregating the precipitation, snow depth, and snowfall measurements within each month and considering the mean value of the minimum and maximum temperature in those intervals. Thereafter, participation is considered as the main measurement, and snowfall, snow depth, and minimum and maximum temperature are exogenous variables. To have an estimate of the auto-regressive and moving average orders, we use the plot of the Partial Autocorrelation Function (PACF) and Autocorrelation Function (ACF) demonstrated in Fig. 6

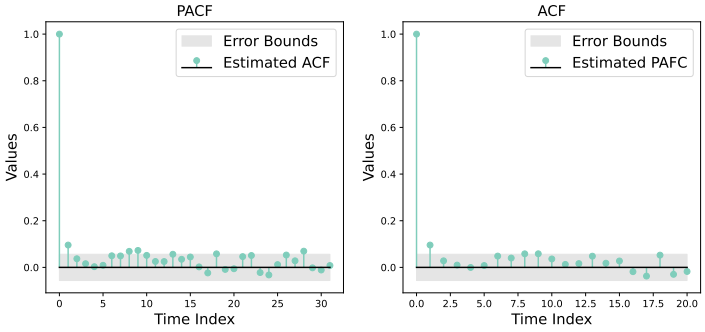

Figure 6: PACF and ACF plots of monthly-aggregated precipitation time series.

As is evident in Fig. 6, the only meaningful lag outside the error bands for both PACF and ACF plots is 1. This points out to the ARMA(1,1) inside the SARIMAX framework. On another note, the small amplitude of this lag also suggests that the time series has only a weak dependence on the latest lag. Based on this observation, we have fitted a SARIMAX$(1, 0, 1)(1, 1, 0, 12)$ on 95% of the data and kept 5% of the data for forecasting. It is sensible to have 6 years of forecasting $(2018 - 2023)$,

which includes 72 next samples. For the trend part, a grid search was performed to obtain the best fit. As is evident, according to the model, the parameters are $p = 1,\ d = 0,\ q = 1,\ P = 1,\ D = 1,\ Q = 0,\ s = 12$. Fig. 7 shows the standardized residuals and Quantile-Quantile (Q-Q) plot of the residual after fitting the observations. As shown in Fig. 7, the Standardized residuals appear to

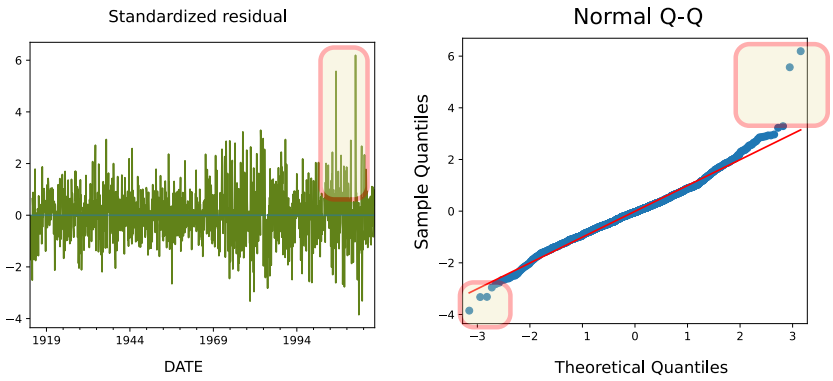

Figure 7: The residuals of the fitted SARIMAX model and the corresponding Q-Q plot.

be normally distributed, as shown in the Q-Q plot. However, the beginning and the ending points deviating or below the tails (Highlighted) indicate heavier or lighter tails than normal distribution. The standardized residuals suggest that these deviations may arise due to the deviation of the data in the last few years, which appears to be two large peaks. That abnormal measurement might be due to short-time but extreme climate change or outliers in the measurement. Table 1 summarizes the calculated coefficients.

Table 1: The coefficients for four exogenous and the AR and MA components of the fitted SARIMAX model.

Measurement	Coefficient
Maximum temperature	-0.5275
Minimum temperature	0.5337
Snowfall	0.0023
Snowdepth	0.0016
AR coefficient	0.2994
MA coefficient	-0.2582

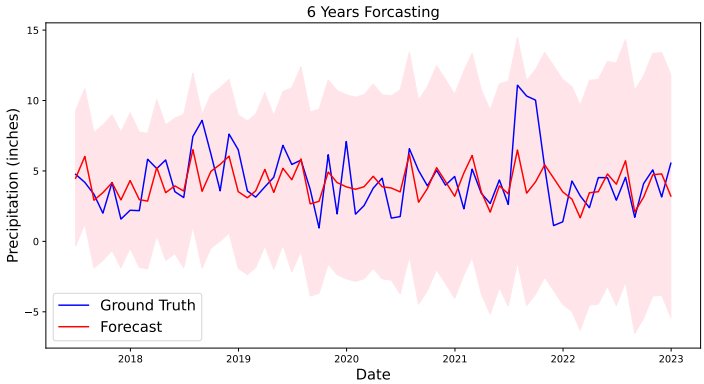

Figure 8: SARIMAX model precipitation forecasting vs. the ground truth in the year 2017-2023.

The values from Table 1 suggest that the maximum temperature is negatively correlated with the precipitation, and on the other hand, the minimum temperature is positively correlated. Additionally, the amount of snowfall and snow depth have insignificant contributions to the prediction of future precipitation. Finally, Fig. 8 demonstrates the forecasting of the future precipitation vs. the ground truth. It can be understood that the model captures the trend up to some extent but fails to have an accurate prediction for each index. Specifically near 2022, where a large peak in the data exists. The mean squared error is 4.01, and the mean absolute percentage error is 42.8%. Additionally, We can see that the further the prediction goes, the wider the confidence interval gets.

Marked Point Process Decoder

Fig. 9 demonstrates the state estimation results for the MPP framework for two consecutive years. According to Fig. 9 (a), the impulses show a better representation of the precipitation sequence as there are some significant impulses that are surrounded by smaller peaks in their vicinity. Fig. 9 (b) shows the decoded state that corresponds well to the existence and the amplitude of the precipitation events. Additionally, it is evident that the state can also be negative; therefore, to have a characterized state recovery, the state is normalized as shown in Fig. 9 (c). Three regions in Fig. 9 are marked to be

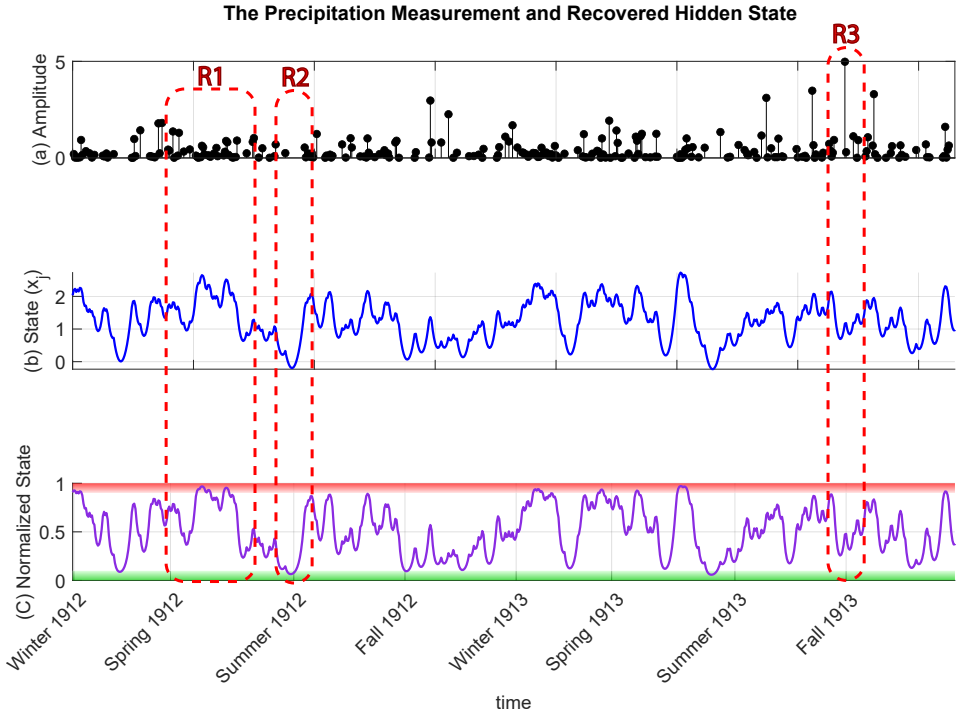

Figure 9: MPP state estimation results for two years: a) the measurement precipitation data, b) The continuous decoded hidden state, c) The normalized decoded hidden state.

discussed. According to region *R1*, the state amplitude is the highest when there is a concentration of the impulses. Such events appear to be recurrent at the end of winter and during the spring season. Meanwhile, the lowest amplitudes (for example, *R2*) occur when there are no precipitation events. Those events are held in the neighborhood between summertime and the end of spring. Another interesting observation is at *R3* where the impulse amplitude is highest. However, the recovered state captures a moderate level.

LSTM predictions and Gaussian process model of residuals

The methodology section outlines the implementation and evaluation of the model, with performance assessed through predictions and residual analysis. Initially, the LSTM model achieves a mean

squared error (MSE) of 0.52. After hyperparameter optimization, the MSE decreases to 0.25. The optimal hyperparameters include a hidden layer size of 215, a single-layer architecture, a dropout rate of 0.23, a learning rate of 0.0123, and a batch size of 16. The LSTM-based precipitation predictions and GP-based residual regression are shown in Fig. 10. For better visualization, the year 1912 is selected to showcase model performance. The red line in the first subplot represents the LSTM predictions, and the black line in the second subplot represents the residuals. The LSTM predictor produces predictions that follow the same trend as the ground truth: high predicted values correspond to periods with multiple impulses, while low predicted values align with periods of no precipitation. This behavior is expected, as learning-based methods tend to focus on the majority and most frequently observed data in the dataset. Impulses, being outliers, are less represented, leading to the model's inability to capture these specific data points. The residuals, dominated by impulses, further support this explanation. By fitting the residuals to a Gaussian Process (GP), uncertainty quantification is achieved. Adding the residuals and their corresponding uncertainty back to the LSTM predictions enhances the predictor, enabling it to generate both "mean" predictions and uncertainty estimates. Specifically, confidence intervals are set as ±1.96 standard deviations of the GP predictions, truncated at zero since precipitation is non-negative.

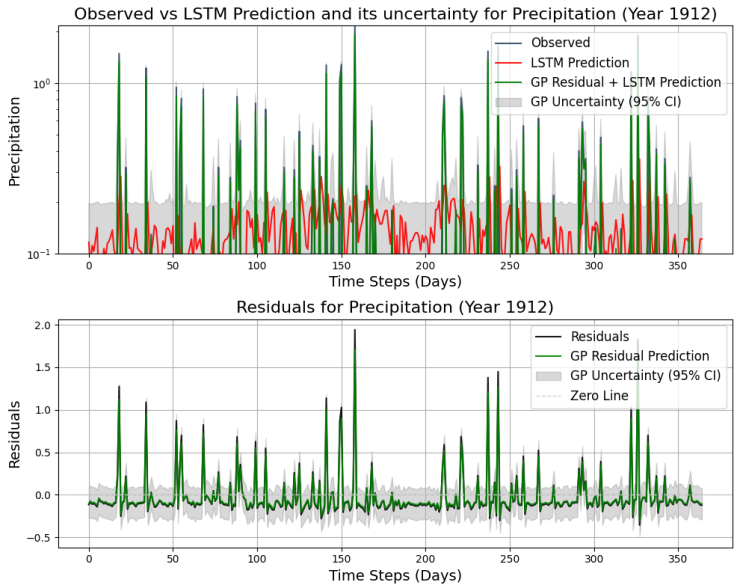

Figure 10: Observed precipitation values, LSTM predictions, LSTM uncertainty band, Gaussian processes of residuals and its mean and confidence interval for the year 1912

Conclusion and discussion

Precipitation prediction has been always a scientific challenge with various applications ranging from real-time flood control to agricultural yield maximization. Physic-based models, probabilistic, and machine learning methods have been used to predict precipitation. This study aims to compare probabilistic methods (such as SARIMAX and MPP) with a deep learning neural network (LSTM). The frequency domain analysis reveals the precipitation data is not periodic and with many impulses, which makes it challenging to generate accurate predictions. In this regard, the MPP approach is utilized to better capture the impulse behavior of the precipitation data. The recovered hidden state gives a proper intuition about the underlying process that leads to rainfall or snowfall events as is evident from the results. This study also evaluated the conventional SARIMAX model for forecasting. While the trend could be captured, the model failed to give an accurate prediction of the impulses (extreme events). One underlying reason could be the Gaussian assumption on the MA process within the SARIMAX structure such that it can't handle well on the out-of-distribution data samples. Another reason could be the fact that the SARIMAX model is based on the linear combination of previous values, errors, and seasonal patterns. Without fine-tuning its parameters, LSTM was unable to capture the trends and absolute values (with 0.52 MSE). After fine-tuning, it was still unable

to predict exact values but could capture the trends moderately (with 0.25 MSE). LSTM has no confidence interval for its predictions, making it hard to use for practitioners who want to use its prediction in real-world operations. Accordingly, GP can be used to provide a reliability estimate of the LSTM prediction. In general, both GP and LSTM are time-consuming to train making it difficult for frameworks that require repetitive training and real-time prediction. The future work includes using the learned MPP model to forecast the future burst of precipitation events. Also, one can think of using heavy tail distribution within a SARIMAX framework to better represent the underlying process. Additionally, constraining the probabilistic or deep learning predictive models with fundamental laws of physics-based models can provide more reliable estimates of precipitation.

References

[1] F. Piadeh, K. Behzadian, and A. M. Alani, "A critical review of real-time modelling of flood forecasting in urban drainage systems," *Journal of Hydrology*, vol. 607, p. 127476, 2022.

[2] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, *et al.*, "Learning skillful medium-range global weather forecasting," *Science*, vol. 382, no. 6677, pp. 1416–1421, 2023.

[3] S. P. Ashok and S. Pekkat, "A systematic quantitative review on the performance of some of the recent short-term rainfall forecasting techniques," *Journal of Water and Climate Change*, vol. 13, no. 8, pp. 3004–3029, 2022.

[4] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*, vol. 38. OUP Oxford, 2012.

[5] D. S. Wickramasuriya and R. T. Faghih, "A marked point process filtering approach for tracking sympathetic arousal from skin conductance," *IEEE Access*, vol. 8, pp. 68499–68513, 2020.

Data availability

Data, code, and report (latex) are available in a GitHub repository: **https://github.com/omidemam/PPTSA.git**

Acknowledgements

The manuscript text has been reviewed using AI-powered proofreading tools.

Omid Emamjomezadeh

Conceptualization, Finding the Dataset and learning to use it, data preprocessing,
methodology, applying an LSTM model, hyper-parameter tuning of the LSTM model, applying Gaussian process to model the residual of the LSTM prediction, analyzing and interpreting the results, presentation preparation, writing the original draft of the report, and editing it.

Ahmadreza Ahmadjou

Implementing the SARIMAX model for forecasting: This includes data processing, training the model, and parameter tuning (try and error).
MPP approach including: Conceptualizing the need to use the model, implementing, analyzing and interpreting the results.
Writing the report including writing and editing the paper and Preparing and presenting presentation preparation.

Ruixuan Zhang

Implementing the frequency domain analysis (DFT and DSTFT); Results analysis and visualization; Report writing and presentation preparation.