**Visualizing Uncertainty with Chromatic Aberration**

Md Rashidul Islam  
Faculty of Computer Science, Dalhousie University   
6050 University Ave, Halifax, Nova Scotia, Canada, B3H 4R2   
md313724@dal.ca

Abstract:  
In recent years an increasing array of research are being conducted by researchers in the field of visualization and in time series forecasting by employing various machine learning models. Since COVID-19 is a respiratory infectious disease caused by novel coronavirus (also known as SARS-CoV-2) and due to its unprecedented challenges over time and global impact, World Health Organization (WHO) has recognized it as a global pandemic. After conducting more than yearlong research, several companies manufactured vaccines with different names and as a result, immunization has started in many countries and that significantly helps to reduce the spread and severity of the infections. From the very beginning scientists and researchers are investigating the perceived data to discover the cause, find the patterns in different countries or demographic areas. So, in this study, we come up with a novel idea for a visualization to present predictive model uncertainties and visualize textures to represent a third property in 2D space. We utilized some common and existing machine learning models to obtain the predicted results and find the model uncertainties for the most impacted countries with respect to number of new-cases, new-deaths, and new-vaccination for different countries. Finally, we visualize the calculated model uncertainties in terms of chromatic aberration and textures in an interactive fashion.

**LIST OF ABBREVIATIONS USED**

D3 - Data Driven Documents

JSON - JavaScript Object Notation

API - Application Programming Interface

ANN - Artificial Neural Network

RNN - Recurrent neural networks

MLP - Multilayer Perceptron

CNN - Convolutional Neural Network

LSTM - Long Short-Term Memory

MAE - Mean Absolute Error

RMSE - Root Mean Square Error

WHO - World Health Organization

**1 Introduction**

The scope of the research includes machine learning models for visualization. Some of the popular predictive machine learning algorithms are used for obtaining the forecasting data and corresponding uncertainty and then visualized with the JavaScript library D3.

**1.1 Background and Motivation**   
The outbreak of coronavirus COVID-19 first emerged in China in December 2019 and the expansion has occurred all over the world and declared as an international public health crisis by the World Health Organization. Since then, the world is quite affected in almost all respects. Various preventive health measures are imposed, and different short-term restrictions are applied to the habitants in different countries at different times. But the mortality rate wasn’t in control significantly until immunization started and over 259 million people have infected and 5.1 million have died in the world over. The infection and death rate are oscillating in different countries due to various reasons such as public insincerity and lack of consciousness about the disease. Moreover, the strain of the virus is changing frequently in different geographical places with more power and variations and a few of the variants like the British variant, the South Africa variant and the Indian variant became the prime concern for the world community, because sufficient research has not been conducted on them yet. Though a great deal of research is being conducted and some vaccines manufactured, and immunization started holistically but still nobody knows when the world will get rid of this cruel pandemic and return to full normal life again.

In recent days, many studies have been conducted to forecast the trend of the spread of the COVID-19 pandemic using various statistical models as well as machine learning models. The autoregressive integrated moving average (ARIMA) model has been widely used in the previous studies to analyze and predict the spread of the disease along with many other popular machine learning algorithms. To our knowledge, most of studies are administered for specific countries. The pandemic started very abruptly and so during the first year, it was difficult to develop any efficient system to forecast anything even for a single day. But after passing more than one year, we have sufficient data to explore, analyze and forecast with the help of modern machine learning algorithms. The ability to identify the expansion rate at which the disease is spreading is very important to confront it and help governments’ regarding contingent policy-making to properly address the consequences of the pandemic and encourage people to follow the rules and health guidelines to achieve the maximum benefit by saving valued lives. The principal objective behind the current research is: to perceive the extent of country-wise uncertainty by discovering property driven predicted results of COVID-19 for certain period of time and employ those uncertainties in visual representation as chromatic aberration that can help the community administrators’ for better planning, providing insights to minimize its impact. Last but not the least the visualization in 2-dimensional space helps the users to perceive dual impacts in single a view. For instance, new cases and new deaths are provided in a single chart with the help of glyph-like texture representations.

**1.2 Basic** **Concepts**To dive into further detail it is very important to introduce thetop-level terms used in the thesis. The following section discusses to raise the basic concept of each of them:

**1.2.1. Uncertainty**

**1.2.2. Texture**

**1.2.3. Chromatic Aberration**

**1.3. Problem statement**   
The primary objective of the research is to present a novel concept of employing chromatic aberration to machine learning model uncertainties by amassing the COVID-19 data. We also present how a tertiary property can be displayed over a two-dimensional viewing system by using the color blended texture elements and convey dual valued message from a single content.

As stated earlier many relevant research are conducted in this domain, so we intend to pursue research by working with machine learning models. For example: CNN is used in some research and MLP is used in other research, so we can keep both in our candidate list.

**1.4. Approach**

At the first step we sought a suitable dataset in terms of completeness for the whole duration of covid. By analyzing numerous data repositories, we came to know that OWID is the most comprehensive one among all others.

Secondly, we had to study different journals about forecasting from temporal data using machine learning models and chose three popular modeling algorithms for our research. Since, finding and comparing the effectives of algorithms’ is out of our scope of work. We randomly chose a reasonable set of the models because we needed to generate the uncertainty data for the countries by using the model predictions.

Thirdly, having the data generation component in python, we needed to write up APIs to connect and pull the data when drawing the charts. Since the model training and data generation for all countries are long running processes, we precompiled the data into json file that can be read and sent back to the client on demand.

Finally, we have chosen D3.js as our front-end library for drawing the charts because it is an efficient platform for visualization prototyping. Since developing the basic drawing algorithms is not our goal, we have fairly taken the benefit of reusing library features but the total data collection, preparation, manipulation, correction and drawing algorithms for various strategic presentations are developed by ourselves.

**1.5. Thesis outline**

TBA in the final paper.

**2. Prior Works:**   
This study involves three major components i. generate time series forecasted data from COVID-19 data using some of the predictive models and calculate corresponding uncertainties for different countries ii. Visualizing Chromatic Aberration to represent the perceived uncertainties in a graphical presentation surface iii. Incorporate data of three variables and orient in a two-dimensional geometric setting. In this section, we are going to include some closely related studies conforming to aspects of the research.

**2.1 Works related to prediction in Machine Learning Models**

On the machine learning forecasting side, Song et. al. [1] compiled monthly data of influenza incidences from all provinces in mainland China from January 2004 to December 2011, comprehensively evaluated and classified these data, and then randomly selected 4 provinces with higher, median and lower incidences, using time series analysis to construct an ARIMA model. The same model but different analysis and forecasting approaches was conducted on the coronavirus disease by other researchers [2]. Recent studies of [3, 4] use Facebook’s Prophet Forecasting Model and ARIMA Forecasting Model to compare their performance and accuracy on the dataset containing the confirmed cases, deaths, and recovered numbers, obtained from the Kaggle website. The forecast models are then compared to the last 2 weeks of the actual data to measure their performance against each other. The result shows that Prophet generally outperforms ARIMA. Several neural network predictive models are used to evaluate their performance against more common machine learning models in a Dengue forecasting project [7]. Srivenkatesh applied Naïve Bayes, logistic regression, support vector machines, Random Forest, K Nearest Neighbour for the examination of liver malady. The classifications are assessed with 5 distinctive execution measurements, i.e., precision, kappa, Mean absolute error (MAE), Root mean square error (RMSE), and F measures. The objective of this query work is to foresee liver infections with different machine learning approaches and pick most efficient algorithm [9]. Results of the examination demonstrated that Logistic Regression classifier demonstrated the best outcomes regarding precision with the least execution times.

**2.2 Uncertainty related works in Visualization**

Botchen et al. [29] focuses on uncertainty that occurs during data acquisition and demonstrates the usefulness of the methods for the example of real-world fluid flow data measured with the particle image velocimetry (PIV) technique. They present two novel texture-based techniques to visualize uncertainty in time-dependent 2D flow fields where in the first method, texture advection is employed to show flow direction by streaklines and convey uncertainty by blurring these streaklines and in a second method isotropic diffusion implemented by Gaussian filtering to continuous change of the density of flow representation.

Error in data is inherent so it cannot be ignored in visualization. Improper or eliminated presentations in visualizations can mislead decision making for data analysts. The goal of uncertainty visualization is to minimize the errors in judgment and represent the information as accurately as possible. This survey Aasim Kamal et al. [30] discusses state-of-the-art approaches such as Quantiﬁcation approach to uncertainty visualization, along with the concept of uncertainty and its sources.

Bonneau et al. [16] explores uncertainty in the visualization domain by comparing different results, such as a weather forecast generated with different parameters and to detect similarities or differences in the results a comparative visualization technique is employed. To compare certain regions in more detail, e.g., borders, they suggested to consider larger comparison areas than individual pixels and it is crucial that data sets which should be compared are visualized next to each other to get a direct comparison for a certain area.

Objective uncertainty of a visual system is evaluated by Barthelme et al. [17] where they discuss the natural perceptual systems involvement with systematic uncertainty because sensory information is imperfect and insufficient to uniquely designate the environment. In their experiment, observers were presented with pairs of images of oriented objects embedded in high levels of noise and had to report the orientation of the image of their choice.

A picture containing text

Description automatically generated

Figure-1: Evaluation of object uncertainty in the visual system (ref. [17])

Figure-1 shows an example of their experiment where they compare objective uncertainty—as computed using the Bayesian framework—with subjective uncertainty, the confidence observers report about their visual perception. To this end, they used a visual task with well-defined statistical properties, discrimination under noise. They report a surprising degree of agreement between objective and subjective uncertainty and discuss possible computational models that could explain this ability of the visual system. Even though the two images contained the same extent of noise, one particular noise structure made an image orientation more obvious than the other. Eventually, observers reliably chose the more obvious of the two images, thereby providing evidence of a capacity to accurately evaluate objective uncertainty.

A statement on the position of uncertainty visualization today is explained in Griethe et al. [18] that defines the basic concept of uncertainty and discusses sources and necessary measures. Visualization is an indispensable approach to the exploration and communication of large data sets of different domains where data sets may contain an unavoidable amount of uncertainty that needs to be included in the visualization process to enable the correct cognition of hidden facts and figures. In addition, it explains how existing approaches could be systematically presented to the acquisition and display of uncertainty can be transferred to new fields, e.g., the visualization of uncertainty in structures.

Uncertainty visualization is a research area that integrates visualization with the study of uncertainty. Among many uncertainties representation of participant-based empirical techniques, there is little evidence in Deitrick et al. [19] to suggest that uncertainty visualization influences in results or decisions. Through a human-subjects experiment, this research evaluates uncertainty visualization methods and indicates that it may affect decisions, but the degree of influence is affected by how the uncertainty is expressed.

State-of-the-art visualization techniques have been successfully engaged in diagnostic medical imaging and Direct Volume Rendering (DVR) sectors and attained maturity in regular clinical works. However, still a major problem is the lack of information on the uncertainty of the tissue classification, which is addressed in the paper Lundstrom et al. [20] by proposing animation methods to convey uncertainty in the rendering. The rendering is animated by sampling the probability domain over time that allows direct user interaction with the classification and it outperforms traditional rendering in terms of assessment accuracy.

Most of the visualization research has ignored the presentation of uncertainty from data because of the inherent difficulty in defining, characterizing and controlling the uncertainty in the visualization process. The paper Pang et al. [21] introduced a wide variety of new uncertainty visualization methods like adding glyphs, adding geometry, modifying attributes, modifying geometry, animation and applied to many applications. The results of the research show that there are a wide variety of possible means to map uncertainty into a scene. The methods presented in the paper represent significant steps toward achieving the goals of uncertainty visualization.

**2.3. Chromatic Aberration history in Visualization**

Again, from a vision perspective, chromatic aberration leads to various forms of color imperfections in the image. When tampering with an image, these aberrations are often disturbed and fail to be consistent across the image. Koh et. al. [10] presented a user study to observe the effect on users’ judgment with Lateral Chromatic Aberration for Chart Reading in Information Visualization on Display Devices and suggested guidelines for information visualization designers to avoid such issues. Yoo et. al. [11] explained Colour illusion on liquid crystal displays and design guidelines for bioinformatics tools to enhance the usability and design of LCD monitors [11]. Lens flare rendering in real-time applications[14] are explained by Hullin et. al. [14] whereas Lee [15] nicely presented the blur effects and focus control to retain a realistic look of the display elements [15]. One of the interesting research projects conducted by Johnson shows that inconsistencies in lateral chromatic aberration can be used to detect tampering in visually plausible forgeries [13].

**2.4. Texture related works in Visualization**

Particle Tracing and Line Integral Convolution (LIC) in Netzel et al. [22] are parallelly and independently used on every pixel of the texture to reduce the computational cost. On top of that a Gaussian low-pass filter with sparse input noise is used for phase shifting along the streamlines. But there is no indication of how high pass filter and/or variable input noise impacts on the result and performance in terms computation and rendering. Streamline computations were replaced by texture advection that works well for both steady and unsteady flow and provides extremely quick results. But the disadvantage of this setup is coupling exponential filter that cannot handle trends properly.

Existing techniques are not capable of accurately aligning and tracking dynamic time-varying data because of the segmentation problem, key feature identification or absence of overlap in consecutive timestep. So, Caban et al. [23] introduces a texture-based feature tracking technique capable of tracking multiple features over time by analyzing local textural properties and finding correspondent properties from synthetic and real-world time varying volumetric data. The main limitation specified in the paper is the cumulative error issue that is caused from the “drifting problem” which exists when small errors are introduced to the texture-based multi-dimensional feature vector over time.

The authors Bachthaler et al. [24] have introduced a new technique of utilising the overlay of two different LIC (line integral convolution) textures to combine the visualization of the tangential and orthogonal vector fields. They have applied a weaving of high-frequency spatial textures of different colors and avoided avoid a direct color blending for compositing. Different filter kernels and filter methods are compared and discussed in terms of visualization quality and speed to obtain a consistent and temporally coherent animation. A perception study was carried out to measure the discrimination and perceived speed of moving patterns under realistic settings. Also, there is an open question to study the implication of global motion perception and the effectiveness of conveying flow structures since they have focused on low-level local motion perception only. The approach of the study is restricted to 2D manifolds and cannot be extended to higher dimensions.

To avoid color blurring and inconsistencies in popular Line Integral Convolution (LIC) scheme and mitigate the expensive computation or memory cost, eliminating surface parameterization, Huang et al. [25] have introduced a novel image-space surface flow visualization approach that preserves the coherence during user interactions. They have employed a precomputed sequence of triangle textures on coordinates of each vertex to ensure noise textures under different viewpoints remain coherent. Although the approach works fine for most models, popping artifacts can be still visible for some complicated models. For example: when the viewpoint is very far away from or very close to the surface.

Kratz et al. [26] have presented a method for the generation of anisotropic sample distributions in the planar and the two-manifold domains. They also presented interactive rendering of anisotropic Voronoi cells. They have used a special sampling approach to generate sample distributions that cover the underlying domain densely while significant holes and cluttered areas are avoided. They use quadratic textures as GPU data structures, which results in some redundant storage that consumes higher memory than it should be required. The most time-consuming step during initial sampling and relaxation in the two-manifold domain is the back-projection. Influence of adding noise to the cell boundaries are not tested in their experiment but have plan to do in future.

To improve the use of color in combination with motion where the author Weiskopf [27] has distinguished between the detection of patterns in motion (seeing the existence) and the actual perception of motion (recognizing speed and direction). It discussed on how calibration is needed to represent data by the perceived speeds of colored patterns and demonstrated how the guidelines of design of animated graphics and the calibration approach can be used. Although they defined and explained the guidelines, they were not able to make a well-established computational model. Finally, they have mentioned several of possible future works, firstly - user studies could be conducted to test the proposed guidelines for various application scenarios, secondly - evaluate the calibration process in more detail by statistically significant user tests, thirdly - address specific combinations of chromatic motion and further perceptual features like texture.

Healey et al. [28] presents a new method for using texture to visualize multidimensional data elements arranged on an underlying three-dimensional height field. Perceptual texture elements are built by controlling three separate texture dimensions: height, density, and regularity. They conducted a set of controlled experiments to measure the effectiveness of these dimensions, and to identify any visual interference that may occur when all three are displayed simultaneously at the same spatial location. Ad-hoc mapping often introduces visual artifacts that actively interfere with a user’s ability to perform their visual analysis tasks. Additionally, it is found that taller, shorter, denser, and sparser pexels can be easily identified, but that certain background texture patterns must be avoided to ensure accurate performance.

**2.5. Limitations of related works**

As stated in the related works section, a plethora of studies have been conducted in these domains, for example: predicting modeling and augmentation of algorithms, time series analyses and comparisons on different diseases and/or on other temporal data, real time predictions from models, measuring chromatic aberration from image distortion, effect of color and light on display devices, uncertainty visualization and decision making, texture analyses and assessments, perceptual textures to represent multi-dimensional dataset, and etc. In our knowledge predictive uncertainty has not been represented with chromatic aberration. Furthermore, our approach of three dynamic variables visualization in two-dimensional space with texture is also a novel idea.

**3 Data Preparation**

Data preparation is one of the most important factors in the research. In the following sub-sections, we explain the raw data and it’s processing to achieve the data for the visualization module.

**3.1.1 Data Collection and Processing**

Data comes bundled in a csv format from ourworldindata.org. The following table shows the list of fields/properties of each record where many of them are not relevant to our research. For example: date, location, new\_cases, total\_cases are some of the useful attributes bolded in the following table.

|  |  |  |
| --- | --- | --- |
| **iso\_code** | continent | **location** |
| **date** | **total\_cases** | **new\_cases** |
| new\_cases\_smoothed | **total\_deaths** | **new\_deaths** |
| new\_deaths\_smoothed | total\_cases\_per\_million | new\_cases\_per\_million |
| new\_cases\_smoothed\_per\_million | population\_density | new\_deaths\_per\_million |
| new\_deaths\_smoothed\_per\_million | stringency\_index | **population** |
| new\_cases\_smoothed\_per\_million | median\_age | aged\_65\_older |
| aged\_70\_older | gdp\_per\_capita | extreme\_poverty |
| cardiovasc\_death\_rate | diabetes\_prevalence | female\_smokers |
| male\_smokers | handwashing\_facilities | hospital\_beds\_per\_thousand |
| life\_expectancy | human\_development\_index |  |

Table-1: COVID Data property list

**3.1.2 Sample Data**

**A picture containing text, appliance

Description automatically generated**

Table-2: screenshot of sample data

In the above Table-2, we have shown only a snapshot of whole dataset where there are hundreds of thousands of records for Covid data for more than 237 countries and territories. Though there are numerous fields in the data, we only needed few of them as listed in previous section. The dataset is collected as a excel file which includes daily occurances and/or counts of all properties. The total\_\* fields like total\_cases, total\_deaths, etc are cumulative and so every day that is updated with previous day’s counts. Data is ordered by date and name of the country correspondingly. If there is no value in a cell for certain date and country then that cell is kept empty, so that is needed to handle during data preprocessing.

**3.2 Machine Learning**

Though we have not done anything novel in machine learning domain, but it is necessary to briefly introduce the salient components that were used in our research to process the available data and generate the required data.

**3.2.1 Definition**

Machine learning is an approach of artificial intelligence (AI)  to provide ability of automatic learning through the use of data. It doesn’t need any explicit programming to perform the task since the algorithms are designed to learn new data intuitively.

**3.2.2 Predictive/Forecasting Models**A time series forecasting model comprises a sequence of data points captured, using time as the input parameter. It uses the historical data to develop a numerical metric and predicts values for the next duration, for instance, data for the next few weeks using that metric.

Forecasting Algorithms

Training Data

New Data

Predictions

After Training

Figure-1: Predictive modeling workflow

**3.2.3 Time Series Analysis vs Forecasting**

Sometimes ambiguity arises between time series analysis with time series forecasting when working with temporal data. As per Shmueli el al. [31] in time series analysis, a time series is modeled to determine its components in terms of seasonal patterns, trends and relation to external factors. In contrast, time series forecasting uses the information in a time series (perhaps with additional information) to forecast future values of that series. The COVID-19 dataset is maintained on a global basis, so it is more trustworthy and with time series forecasting models can be considered as suitable for our research to get the predicted results and hence generate our required uncertainty data to represent chromatic aberration in visualization area.

**3.2.4 Concerns of Forecasting**Time series forecasting is an important area of machine learning. It is important because there are so many prediction problems that involve real life issues like time component. In forecasting it is very important to understand the goal of the problem and the nature of the available data. For instance, the volume of data, time horizons (short, medium or long term), frequency of update etc. plays an important role in forecasting. Sometimes time series data requires cleaning, scaling and even transformation, for example: if there are gaps/missing data, if there are outliers or corrupt data then those need to be addressed. Depending on the frequency, a time series can be of yearly (e.g., annual budget), quarterly (e.g., expenses), monthly (e.g., air traffic), weekly (e.g., sales quantity), daily (e.g., weather forecast), hourly (e.g., stock market price), minutes (e.g., inbound calls in a call canter) and even seconds wise (e.g., web traffic). Being the covid pandemic world-wide concerns for whole humanity, we use the daily forecast mechanism to our research. To compare the results side by side we have created prediction for 200 days from every models.

**3.3 MLP**  
A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). It is a neural network connecting multiple layers in a directed graph, which means that the signal passes through the nodes only in one direction. It can be used for time series forecasting by taking multiple observations at prior time steps, called lag observations, and using them as input features and predicting one or more time steps from those observations. The training dataset is therefore a list of samples, where each sample has some number of observations from days prior to the time being forecasted, and the forecast is the next days in the sequence.

Diagram

Description automatically generated

Figure-2: Basic Architecture of MLP network [ref. 33]

We use the rectified linear activation function on the hidden layer as it performs well and a linear activation function on the output layer because we are predicting a continuous value. We use root squared error as loss function and the ‘adam’ optimizer for training the network.

The following steps shows the algorithm used to setup our MLP model:

---------------------------------------------------------------------------------------------------------------------

1. Take an instance of ‘Sequential’ Model from Keras deep learning library.
2. Add a Dense layer to the model with stating number of inputs (24), number of nodes (500), number of epochs (100) and batch size (100), rectified linear activation function (relu).
3. Add another Dense layer with number of outputs (1), since we predict a continuous value.
4. Compile the model with mean square error (mse) loss function and ‘adam’ optimizer.
5. Fit the model with training data set for number of epochs (100) and batch size (100).
6. Make an ensemble of models by following the steps 1 to 5.
7. Get prediction ‘yhat’ for each time step (day) from all the models of the ensemble.
8. Calculate the ranges (lower level, mean and upper level) of each prediction.
9. Calculate uncertainty of the model for each day by using the set of yhats using the uncertainty calculating formula explained in 3.7.

---------------------------------------------------------------------------------------------------------------------

Algorithm-1: MLP Model

**3.4 CNN**Convolutional Neural Networks are a type of deep neural network developed for computer vision; for instance, two-dimensional image data, although they can be used for one-dimensional data such as sequences of text and time series forecasting. When operating on one-dimensional data, the CNN reads across a sequence of lag observations and learns to extract features that are relevant for making a prediction.

Diagram

Description automatically generated

Figure-3: Basic Architecture of CNN network [ref. 34]

We define a CNN with two convolutional layers, one max-pooling layer, one flatten layer, and a dense layer from the input sequences. 1D convolution (Conv1D) layer (e.g., temporal convolution creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs. They have a configurable number of filters, kernel size, pool size and rectified linear activation function is used as loss function. The number of filters determines the number of parallel fields on which the weighted inputs are read and projected. A max pooling layer is used after convolutional layers to distill the weighted input features into those that are most salient, reducing the input size by 1/2. The pooled inputs are flattened to generate a long vector before being interpreted and used to make the prediction.

To dive into further we need to briefly introduce some of the basic terms used in this model:

**Conv1D:**

A convolution layer transforms the input image in order to extract features from it. This layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs

**MaxPooling1D:**

Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

The following steps shows the algorithm used to setup our CNN model:

---------------------------------------------------------------------------------------------------------------------

1. Take an instance of ‘Sequential’ Model from Keras deep learning library.
2. Add a Conv1D layer to the model defining the number of filters (24), kernel size (500), input shape (100), rectified linear activation function (relu).
3. Add another Conv1D layer with same settings but without input shape.
4. Add another MaxPooling1D layer with pool size of 2.
5. Flatten (reshape) the result of previous step into single dimension before interpreted by the next layer.
6. Add a Dense layer with number of outputs (1), since we predict a continuous value.
7. Compile the model with mean square error (mse) loss function and ‘adam’ optimizer.
8. Fit the model with training data set for number of epochs (100) and batch size (100).
9. Create an ensemble of 6 models by following the steps 1 to 8.
10. Get prediction ‘yhat’ for each time step (day) from all the models of the ensemble.
11. Calculate the ranges (lower level, mean and upper level) of each prediction.
12. Calculate uncertainty of the model for each day by using the set of yhats using the uncertainty calculating formula explained in 3.7.

---------------------------------------------------------------------------------------------------------------------

Algorithm-1: CNN Model

**3.5 LSTM**

The LSTM neural network is a member of RNN and it can be used for univariate time series forecasting. It uses an output of the network from a prior step as an input in attempt to automatically learn across sequence data. The LSTM has an internal memory allowing it to accumulate internal state as it reads across the steps of a given input sequence.

**Diagram

Description automatically generated**

Figure-4: Basic Architecture of LSTM network (ref. 35)

For this model we define a LSTM layer from inputs and subsequently two dense layers. Like other models, rectified linear activation function is used in LSTM layer and in one of dense layer. A simple grid search of model hyperparameters was performed with the predefined configuration.

The following steps shows the algorithm used to setup our LSTM model:

---------------------------------------------------------------------------------------------------------------------

1. Take an instance of ‘Sequential’ Model from Keras deep learning library.
2. Add an LSTM layer to the model defining the number of nodes (24), input shape (100), rectified linear activation function (relu).
3. Add a Dense layer for 24 input nodes and ‘relu’ activation function.
4. As we predict single value output, add a Dense output layer of 1 node.
5. Compile the model with mean square error (mse) loss function and ‘adam’ optimizer.
6. Fit the model with training dataset for number of epochs (100) and batch size (100).
7. Create an ensemble of 6 models by following the steps 1 to 8.
8. Get prediction ‘yhat’ for each time step (day) from all the models of the ensemble.
9. Calculate the ranges (lower level, mean and upper level) of each prediction.
10. Calculate uncertainty of the model for each day by using the set of yhats using the uncertainty calculating formula explained in 3.7.

---------------------------------------------------------------------------------------------------------------------

Algorithm-1: LSTM Model

TODO:

1. Algorithm for each chart
2. Algorithm for texture representation
3. Algorithm for color assignment of textures

**3.6 ARIMA**  
An Autoregressive Integrated Moving Average (ARIMA), is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. A statistical model is autoregressive if it predicts future values based on past values. It is a very popular technique for time series modeling. It describes the correlation between data points and considers the difference of the values. ARIMA models work better with the following assumptions –

* The data series is stationary, which means that the mean and variance should not vary with time. A series can be made stationary by using log transformation or differencing the series.
* The data provided as input must be a univariate series since it uses the past values to predict the future values.

The model has three major components which come from its name – AR (autoregressive term), I (Integrated term) and MA (moving average term). Let us try to briefly explain each of these components –

* AR term refers to predicting the next value using the prior values of dataset. The AR term is defined by the parameter ‘p’ in ARIMA.
* Integrated(I) term represents the number of times the differencing operation is performed on series to make it stationary (i.e., data values are replaced by the difference between the data values and the previous values). Test like ADF can be used to determine whether the series is stationary and help in identifying the d value. Differencing is only needed if the series is non-stationary otherwise, no differencing is needed, and in that case d=0.
* MA term is used to define number of prior/lagged forecast errors used to predict the future values. The parameter ‘q’ in ARIMA represents the MA term.

**3.6.1 Auto ARIMA**Although ARIMA is a very powerful model for forecasting time series data, the data preparation and parameter tuning processes end up being really time consuming. Before implementing ARIMA, it needs to make the series stationary, and determine the values of p and q as stated earlier. Auto ARIMA makes this complicated task simple for us as it eliminates those time-consuming tasks. Below are the steps you should follow for implementing auto ARIMA:

---------------------------------------------------------------------------------------------------------------------

1. Load data: Collect data from the source repository and load into a data table.
2. Preprocess data: As the prerequisite of the model input is to be univariate, drop other columns from the data table and make sure all empty values with NULL so that system does not break during runtime.
3. Fit Auto ARIMA: Fit the model on the univariate series of data
4. Predict values: Make predictions on the validation set by using the prior values.
5. Calculate Series: Calculate series by using the forecasted results in earlier step.
6. Find the lower and upper bound of the series which will be used to calculate the uncertainties of the prediction.

---------------------------------------------------------------------------------------------------------------------Algorithm: ARIMA Model

**3.7 Uncertainty Data Generation**

Uncertainties are calculated from the actual and predicted values. The current calculation is very straight forward where we find the difference between actual and predicted values. Then find the maximum difference to set out the domain of the difference. Finally, divide each difference by maximum difference and multiply by a scaling factor to keep the maximum result in single digit. Here is given the steps to find the uncertainties using the machine learning models:

1. Read data from filesystem (excel file) to Data-Frame
2. Select Fields for which we need to generate uncertainty data
3. Create Machine Learning model for MLP/CNN/LSTM
4. Split data into training and test set
5. Train model with training set
6. Use model to get predicted or forecasted results
7. Find uncertainties or prediction error from model
8. Continue step 3 to 7 for each field and each model
9. Store uncertainty data as json in filesystem

Algorithm-1: calculate uncertainty using predictive models

**3.7.1 Uncertainty Data Scaling**

We have shown top-level algorithm in the above section to generate uncertainty data. Since the uncertainty values are pretty larger to accommodate in display, so it needed to scale in certain level. The following pseudo code is used to scale the uncertainty data.

1. country\_avg\_error = pred\_errors\_of\_all\_dates/number\_of\_days
2. max\_error = find\_max\_error(all\_country\_avg\_errors)
3. scaling\_factor = 7
4. country\_uncertainty = country\_avg\_error \* scaling\_factor / max\_error;

Algorithm-2: data scaling

**3.7.2 Snapshot of uncertainty data**

Since the pandemic affected all the countries of the world and there are more than 200 countries, so we have trained the models for top 100 countries which were infected severely. Based on that setup, we have sorted the countries by obtained uncertainties in both ascending and descending orders. The following two tables shows the top 10 uncertainty attaining countries and the bottom one shows the lowest 10 uncertainty attaining countries.

**3.7.3 Top 10 uncertainty countries using MLP model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Country** | **Actual Count** | **Predicted Count** | **Uncertainty** |
| United States | 14,851,118 | 15,652,300 | 7.00 |
| India | 15,693,425 | 7,409,636 | 4.28 |
| Brazil | 7,219,982 | 7,409,636 | 3.64 |
| Kazakhstan | 667,009 | 651,009 | 2.43 |
| France | 2,088,610 | 2,307,005 | 2.15 |
| Peru | 432,034 | 546,901 | 1.28 |
| Germany | 1,700,161 | 1,599,684 | 1.21 |
| Spain | 1,542,012 | 1,510,467 | 1.07 |
| Turkey | 3,645,288 | 3,389,016 | 1.03 |
| Argentina | 2,352,216 | 2,450,255 | 1.02 |

Table-3: Top uncertainty countries in the ordered list

**3.7.4 Lowest 10 uncertainty countries using MLP model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Country** | **Actual Count** | **Predicted Count** | **Uncertainty** |
| Qatar | 36,256 | 36,796 | 0.013 |
| Albania | 62,292 | 65,515 | 0.016 |
| Estonia | 90,950 | 89,900 | 0.017 |
| Egypt | 118,376 | 124,175 | 0.019 |
| Moldova | 103,270 | 101,832 | 0.019 |
| Australia | 161,819 | 147,134 | 0.021 |
| Algeria | 86,238 | 82,121 | 0.022 |
| Singapore | 178,151 | 175,400 | 0.025 |
| North Macedonia | 57,447 | 57,420 | 0.037 |
| South Korea | 277,584 | 274,766 | 0.037 |

Table-3: Lowest uncertainty countries in the ordered list

From the above two tables, it is clearly noticeable that uncertainty is completely independent on the number of cases (Actual Count). For example: United States has lower number of cases than India but achieved higher uncertainty than India. Again, Kazakhstan and France exhibit same behavior and if we examine other countries then surely, we will get more.

**3.7.5 Uncertainty Comparison among Models**

|  |  |  |  |
| --- | --- | --- | --- |
| **Country** | **MLP** | **CNN** | **LSTM** |
| United States | 7.00 | 7.00 | 3.44 |
| India | 4.28 | 0.61 | 7.00 |
| Brazil | 3.64 | 0.51 | 3.24 |
| Kazakhstan | 2.43 | 0.42 | 0.35 |
| France | 2.15 | 0.31 | 0.81 |
| Peru | 1.28 | 0.23 | 0.28 |
| Germany | 1.21 | 0.19 | 0.50 |
| Spain | 1.07 | 0.19 | 0.67 |
| Turkey | 1.03 | 0.19 | 1.21 |
| Argentina | 1.02 | 0.14 | 1.08 |

From the above comparison table of three different machine learning models, we notice that the uncertainties greatly vary for each country based on the model. There is no country which has identical uncertainty values for all three models. Though the dataset used in each of the models in similar approach, the variation appears due to their internal mechanism of the model algorithms. Since the model superiority examination is not our goal, we are not going to discuss further about it. We use the uncertainty data whatever we obtained from model prediction and uncertainty calculation methods.

**4 Visualization**

As stated earlier we generate required data from the standalone python program with the predictive models and save it in JSON format in file system. Then for the front-end we pull it through an API call and feed in client-side scripts for drawing charts since chart presentation is the key part of our research.

Describe about Streamgraph here

**4.1 Web Interface**

To visualize different charts, we have developed a web-interface with several html input controls in the top toolbar and all charts are presented in the main container placed just below the toolbar.  
 **Chart

Description automatically generated with medium confidence**

Figure-5: Initial Web Interface (Left - Bubble chart, right – Streamgraph)

In the following section, we briefly explain the basic functionalities of the input fields in toolbar.

**Chart dropdown**: List of chart names, on selection it will automatically draw the corresponding chart in the main container.

**Select Model:** Names of the predictive models for which we have generated data for charting.

**Country stream with**: We have an option to draw stream graph by using the actual property data or with the predicted data.

**Reset:** Return to the initial state of the settings.

**Pan Chart:** Since the bubble chart and stream graphare drawn side by and they work interactively like filtering the streamgraph with the selection from bubble chart, so sometimes it is necessary to zoom-in/out of the charts and consequently panning the charts in its own space is also advantageous.

**Wing Stream:** changes the drawing mode to interact with mouse events. We explain later how wing stream works.

**Bubbles Select:** Select the country from the chart and redraw with the selected ones only.

**Bubbles Remove**: Filter-out countries from the bubble chart. In this mode the selected countries are omitted from the chart. After omitting countries on press ‘Go’ button it redraws with the other countries.

**Stream Graph:** By default, the system shows the stream graph at the right side of the container and bubble chart at the left. The stream graph is filled with different colors for every country.

**Reshuffle Streams:** Allows to draw main streamgraph with the selected countries of interest from bubble chart.

**Texture Stream:** This is a toggle button to switch the stream graph from color-based filling to texture based filling, that means instead of flat color flow it uses bullet like textures to fill the stream but still they have different colors for their own country region. More detail is shown in section 4.5.

**Tertiary Property**: Toggles to impose third property in right side of stream graph. That means it helps to add a third property like uncertainty over the same stream graph and see the effect.

**4.2 Filtering**

We use data for top 100 countries based on the total infection rate. As we see from the Figure-5, it is difficult to read the label of the country and difficult to identify the extent of aberration for the smaller circles having lower uncertainties. That’s why we implement a filtering option with different perspectives. In the section below we briefly explain them.

**4.2.1. With Selective Countries**

**Chart, bubble chart

Description automatically generated Chart, bubble chart

Description automatically generated** Figure-6: With selected countries of interest

In this mode, it allows users to select the countries of interest on first click and toggles on the next one. So, when all preferred countries are selected the ‘Go’ button redraws the bubbles side by side with comparatively bigger sizes.

**4.2.2 By removing countries**

**Chart, shape, bubble chart

Description automatically generated A picture containing honeycomb, outdoor object

Description automatically generated**

Figure-7: Removal of countries of interest

This is the opposite of the earlier one where the user can select the countries to remove from the chart, for instance, removing bigger ones help to find the status of the countries having a smaller size.

**4.4 Main Streamgraph**

**A picture containing background pattern

Description automatically generated Background pattern

Description automatically generated**

Figure-8: Reshuffling main streamgraph

In this mode, a user can choose the countries from the bubble chart. On select the countries, the corresponding ones will be highlighted in the streamgraph to represent the selection and the rest of the country-streams will be grayed out in the same chart. To confirm the execution streamgraph redraws with the selected countries as shown in the Figure-8.

**4.5 Display Tertiary Property with Texture**

As said, our approach of visualizing additional property in two-dimensional space is novel because similar works were done in different approach such as Healey et al [28]. For example: we show the stream of one property like ‘New Cases’ of all countries vs date. Since we are working in two-dimensional space so it’s a challenge of showing another property like ‘Death Rate’ on top of it. So, texture can be a good option to serve our purpose.

**Background pattern

Description automatically generated Background pattern

Description automatically generated**

y

x

Hotspot2

Hotspot1

Figure-9(A): Realization of third property effect



New cases



date

0%

50%

100%

80%



Death Rate by fill percentage



Figure-9(B): Three properties representation

Now the question is how does it work? To answer this question, we have marked two hotspots in the above Figure-9(A) where we see in hotspot1 the smaller circles are much closer, but for hotspot2 this is changing a lot. This is happening because for hotspot1 the death rate is much higher than the hotspot2.

In Figure-9(B) it gives indication of how three properties work in the 2D space where ‘Date’ and ‘New Cases’ represents traditional (x, y) axes respectively and ‘Death Rate’ is represented by the percentage of fill of the circle.

**4.6 Star Fish Inspiration**In this approach, user can draw multiple stream graphs by dynamically calculating the position of the country cell and its corresponding start point in the cell center and angle to place it without (or possibly minimum) overlapping the other countries’ streams. If we call each individual stream as a wing, then the benefit of this chart is it allows one to draw many charts in compact way. Another interesting feature is that we accommodate multiple properties in each stream, for example: mlp, cnn and lstm predictions are used in bottom chart whereas total\_cases, new\_cases, new\_deaths, icu\_patients, hosp\_patients, new\_tests are used for top chart.

A picture containing wheel, gear

Description automatically generatedA picture containing wheel, gear

Description automatically generated

Figure-10: Approach to represent multi-country-actual counts(left), predicted counts(right)

**4.7 Parallel Coordinates Chart**

Parallel plots or parallel coordinates plots allows one to compare the features of several individual observations (series) on a set of numeric variables. Each horizontal axis represents a variable and often has its own scale and the units can even be different, that is the strength of this special kind of plots. The main advantage offered by parallel coordinate is the representation of high dimensional data as a 2-dimensional visualization. Data is represented in the form of a polyline, and it becomes possible to perceive trends shown by data entries from the visualization.

Diagram

Description automatically generated

Figure-11: Parallel coordinates chart

This plot is helpful in our presentation because we have several variables together to visualize one after another and showing the relationships between them. For example, you can compare number of total cases(total\_cases) with hospitalized patients (hosp\_patients) facilitated by a tooltip showing the country name. Also, it can show the predicted flow (thinner line) along with actual counts (thicker line). The limitation of this chart is frequent occlusions.

**4.8 Impact Chart**

This chart helps to indicate daily uncertainty presentation for every country as a cell. In this way a user can perceive trends for certain day or a set of consecutive days. In other words, the chart provides a useful platform that helps you decide which uncertainty requires your attention. So, if this tool was used by WHO then the administrator could consider which countries are vulnerable tomorrow or the day after tomorrow.

A picture containing text

Description automatically generated

Figure-12: Impact chart

**4.9 Horizontal Chart**

Horizontal charts are small-multiple area charts that allow greater precision for a given vertical space by using colored bands. These charts can also be used with diverging color scales to differentiate positive and negative values.

Background pattern

Description automatically generated

Figure-13: Horizontal chart

**4.10 Usage Chart [Work in progress]**

This chart is more much like impact chart because their construction style is mostly similar to each other, though the axes are used in reverse order. Need to brainstorm on it about how this chart can be used in our research in meaningful way.

Chart, histogram

Description automatically generated

Background pattern

Description automatically generated

Figure-14: Charts of Daily counts (yet to impose uncertainty)

**4.11 Chromatic Aberration Example**

Chromatic aberration is term in optics where it refers to distortion and spherochromatism that happens due to the failure of a lens to focus all colors to the same point. In other word, chromatic aberration is a color distortion that creates an outline of unwanted color along the edges of objects in a photograph.

**A picture containing plant, tree

Description automatically generated A frisbee flying over a bridge

Description automatically generated with low confidence  
Figure - Example of chromatic aberration (collected)**

**4.11.1 Implementation Mechanism**

**Diagram, engineering drawing

Description automatically generated Shape, circle

Description automatically generated**

(x,y)

Figure- Geometric concept(left), Implementation with a circle(right)

To draw a circle representing aberration, we draw 3 circles internally, let’s call them 3 chromatic circles. The following technique is applied on each of the chromatic circles -

* Once for color (r, 0, 0) with a shifted location of (x, y + r)
* Once for color (0, g, 0) with a shifted location of (, )
* Once for color (0, 0, b) with a shifted location of (, )

Where ‘r’ is the radial offset of each of the 3 circles from the center of the original circle located at (x, y).

By using the above formula, a resultant aberration is presented with the uncertainty for the country France (FRA) in the above figure(right). Though in real pictures, we found the aberration is shown as a kind of blurring or fading but here we present one with equal intensity highlighted color though the concept remains the same.

**5 Discussion**TBA in the final paper.

**5.1. Known issues to work out**

The following issues are yet pending to implement

- Finish ‘Usage Chart’ by finding a way to inject aberration information on each slice of the day/country.

- Improve aberration of ‘Impact Chart’ chart in terms of color matching for count part with the uncertainty part.

- Uncertainties are currently calculated based on the difference between predicted values and actual values. Try with more options like MAE, RMSE, Accuracy, F1 Score, etc. and check which option gives better result in terms of visualization.

**5.2. Limitation**

There are several issues in our proposed solution of chromatic aberration. For example: in real aberration in picture the blurring happens very slowly from inner edge to outer edge but in our case, it just gives us a range of uncertainty for the prediction, so the whole edges are with bright color. However, our simplified implementation allows us to reduce the aberration to a single parameter, which facilitates chromatic aberration tuning with regards to the amount of represented uncertainty.

In texture presentation we have generated texture patterns with linear gradient so the color intensity in left of the bullet point higher than the right side. So, it is an open problem to improve and ensure the intensity of the color for the visible part of the circular textures.

**5.3. Future Work**

TBA in the final paper.

From Prof. Mayra/Brooks-  
And note these for future work:

1. Is it possible to have different hues of chromatic aberration? If yes, another possible study can be which CA hue works better.
2. When comparing the CA to other alternatives, you can use eye-tracking to get qualitative data.

… the 2nd one because with Covid we will be doing an online only study, so we don’t be able to use our eye tracking system.

**References:**

[1] Song, Xin; Xiao, Jun PhD; Deng, Jiang PhD; Kang, et al. Time series analysis of influenza incidence in Chinese provinces from 2004 to 2011. Received March 1, 2016, Accepted May 20, 2016, Medicine: June 2016 - Volume 95 - Issue 26 - p e3929

[2] Muhammad Ali, Dost Muhammad Khan, et al. Forecasting COVID-19 in Pakistan, received: August 17, 2020; Accepted: November 10, 2020; Published: November 30, 2020.

[3] COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population. Appl. Sci. 2020, 10(11), 3880; Received: 5 May 2020 / Revised: 23 May 2020 / Accepted: 29 May 2020 / Published: 3 June 2020

[4] Christophorus Beneditto, Aditya Satrio et al. Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET, https://doi.org/10.1016/j.procs.2021.01.036  
  
[5] Leo J, Luhanga E, Michael K. Machine Learning Model for Imbalanced Cholera Dataset in Tanzania. The Scientific World Journal. 2019 Jul; 2019: p. 1–12.

[6] Emrah Gecili, Assem Ziady, Rhonda D. Szczesniak. Forecasting COVID-19 confirmed cases, deaths and recoveries: Revisiting established time series modeling through novel applications for the USA and Italy. **Received:** June 30, 2020; **Accepted:** December 5, 2020; **Published:** January 7, 2021.  
  
[7] Sathler C, Luciano J. Predictive modeling of dengue fever epidemics: A Neural Network Approach. 2017. Data Science for Drug Discovery, Health and Translational Medicine. December 10, 2017. I590.

[8] Miranda GHB, Baetens JM, Bossuyt N, Bruno OM, Baets BD. Real-time prediction of influenza outbreaks in Belgium. Epidemics. 2019 Sep; 28: p. 100341.

[9] Muktevi Srivenkatesh, Performance Evolution of Different Machine Learning Algorithms for Prediction of Liver Disease. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-2, December 2019.

[10]        K. Koh, B. Kim & J. Seo. 2014. Effect of lateral chromatic aberration for chart reading in information visualization on display devices. Advanced Visual Interfaces. Como, Italy, 289-292.

[11]        H. S. Yoo. 2007. Color illusions on liquid crystal displays and design guidelines for information visualization. Master of Science, Virginia Tech.

[12]        T. Boult & W. Wolberg. 1992. Correcting chromatic aberrations using image warping. CVPR, Champaign, IL, 684–87.

[13]        M. K. Johnson & H. Farid. 2006. Exposing digital forgeries through chromatic aberration. Multimedia and security, Geneva, Switzerland, 48-55.

[14]        M. Hullin, E. Eisemann H.P. Seidel & S. Lee. 2011. Physically-based real-time lens flare rendering. ACM SIGGRAPH, Vancouver, 108:1–108:9.

[15]        S. Lee, E. Eisemann & H.P. Seidel. 2010. Real-time lens blur effects and focus control. ACM SIGGRAPH, Los Angeles, 1-7.

[16] Bonneau et al. Overview and State-of-the-Art of Uncertainty Visualization, The University of Grenoble, France e-mail: Georges-Pierre.Bonneau@ujf-grenoble. fr. ISBN: 978-1-4471-6496-8

[17] Simon Barthelme, Pascal Mamassian. Evaluation of Objective Uncertainty in the Visual System. Received June 8, 2009; Accepted August 12, 2009; Published September 11, 2009.

[18] Henning Griethe et al. The Visualization of Uncertain Data: Methods and Problems. Computer Graphics, 18051 Rostock, Germany. January 2006, 2,988 reads, 36 publications, 262 citations.

[19] Deitrick, S., Edsall, R.: The influence of uncertainty visualization on decision making: An empirical evaluation. In: Progress in Spatial Data Handling, pp. 719–738. Springer Berlin Heidelberg (2006).

[20] Lundstr¨om, C., Ljung, P., Persson, A., Ynnerman, A.: Uncertainty visualization in medical volume rendering using probabilistic animation. IEEE Transactions on Visualization and Computer Graphics 13(6), 1648–1655 (2007).

[21] Pang, A., Wittenbrink, C., Lodha., S.: Approaches to uncertainty visualization. The Visual Computer 13(8), 370–390 (1997).

[22] Rudolf Netzel and Daniel Weiskopf, Texture Based Flow VisualizaTion. November 2013, Computing in Science and Engineering 15(6): 96-102,

[23] Jesus J. Caban, Alark Joshi, and Penny Rheingans. Texture-based feature tracking for effective time-varying data visualization, IEEE Transactions on Visualization and Computer Graphics (Volume: 13, Issue: 6, Nov.-Dec. 2007). **Page(s):**1472 – 1479.

[24] Sven Bachthaler, Daniel Weiskopf. Animation of Orthogonal Texture Patterns for Vector Field Visualization. IEEE Transactions on Visualization and Computer Graphics (Volume: 14, Issue: 4, July-Aug. 2008), **Page(s):**741 – 755.

[25] Jin Huang, Zherong Pan, Guoning Chen, Wei Chen, Hujun Bao. Image-Space Texture-Based Output-Coherent Surface Flow Visualization. IEEE Transactions on Visualization and Computer Graphics ( Volume: 19, Issue: 9, Sept. 2013). **Page(s):**1476 – 1487.

[26] Andrea Kratz, Daniel Baum, and Ingrid Hotz. Anisotropic Sampling of Planar and Two-Manifold Domains for Texture Generation and Glyph Distribution. IEEE Transactions on Visualization and Computer Graphics ( Volume: 19, Issue: 11, Nov. 2013). **Page(s):**1782 – 1794.

[27] D. Weiskopf. On the role of color in the perception of motion in animated visualizations. **Conference:**10-15 Oct. 2004, Austin, TX, USA. IEEE Visualization 2004. **ISBN:** 0-7803-8788-0.

[28] C.G. Healey; J.T. Enns. Building perceptual textures to visualize multidimensional datasets. 18-23 Oct. Research Triangle Park, NC, USA. 1998. Proceedings Visualization '98 (Cat. No.98CB36276). **ISBN:** 0-8186-9176-X**.**

[29] R.P. Botchen; D. Weiskopf; T. Ertl. Texture-based visualization of uncertainty in flow   
fields. VIS 05. IEEE Visualization, Minneapolis, MN, USA. 23-28 Oct. 2005. **ISBN:** 0-7803-9462-3.

[30] Aasim Kamal · Parashar Dhakal, et al. Recent advances and challenges in uncertainty visualization: a survey. May 2021, Journal of Visualization 24(5):1-30.

[31] Galit Shmueli, Kenneth C. Lichtendahl Jr. Practical Time Series Forecasting with R: A Hands-On Guide [2nd Edition] (Practical Analytics) Paperback – July 19, 2016. Page 18-19. ISBN-13 978-0997847918

[32] Jason Brownlee. Deep Learning Models for Univariate Time Series Forecasting. https://machinelearningmastery.com/how-to-develop-deep-learning-models-for-univariate-time-series-forecasting.

# [33] Aayush Agrawal, Building Neural Network from scratch. https://towardsdatascience. com/building-neural-network-from-scratch-9c88535bf8e9

[34] Akinori Hidaka, Takio Kurita. Consecutive Dimensionality Reduction by Canonical   
Correlation Analysis for Visualization of Convolutional Neural Networks. Conference: Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications. December 2017. Pages 160 – 167.

[35] Z. Wang et al. Model identiﬁcation of reduced order

using deep learning. International Journal for Numerical Methods in Fluids. Int. J. Numer.