Infectious Disease Risk Predictor

118

Team Leader : Nathanael Ahiagbedey (*CE/CS*)
Team Members : Connor Van Etten (*CS*), Danny DeJesus (*CE/CS*), Zavier Romano (*CS*)
Faculty Advisor : Dr. Sunnie Chung
Electrical Engineering and Computer Science, Cleveland State University, **Cleveland, OH 44108**
m.ahiagbedey@vikes.csuohio.edu

Abstract

Goal
To develop a robust system which can accurately display and predict both current and future disease trends and patterns by applying a **Time Series Forecast** to analyze large scale datasets of CDC data tables.

Design Objectives
- Collection and processing of **Big Data** from accredited sources within the medical reporting space.
- Implementing an **automated collection pipeline** for disease information with python to our hosted MySQL database server.
- Retrieving processed datasets from our Database and performing intelligent predictive analysis on them, before returning result to our remote database server
- Setting up a efficient cloud environment to host our Database and Web Application so our project is not only stable but scalable.
- Design an interface to help users make accurate and easy decisions regarding public health.

Introduction and Background

Current Trend with Publicly Available Disease Data
- Multiple streams of data with conflicting views.
- Accuracy varies greatly regarding source of data.
- Reputable sources make data hard to digest.
- Most publicly usable data normally comes from Social Media.

Challenges regarding Big Data
- Minimal formatting, needing a large amount of preprocessing to make useful.
- Limited trustworthy sources and APIs.

System Design

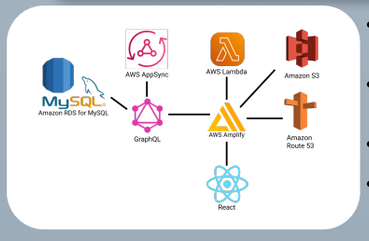

Fig 1 : Cloud Environment Architecture

- Cloud Infrastructure was hosted in Amazon Web Services (**AWS**) making it highly scalable for the future.
- Data collection and Machine learning currently are hosted on local machines and use **CI/CD** pipelines to impact change to our remote servers
- Tech stack for Web Development includes **Node.js**, **React**, **Chart.js**, **D3**, and **Framer-Motion**.
- Tech Stack for machine learning was built includingstate-of-the-art machine learning libraries such as **TensorFlow**, **NumPy** and **Pandas**.

Web Application User Interface Design

- Application frontend was created using React.js framework.
- Displays disease information using Chart.js to give visual graphics to showcase predictions and current trends.

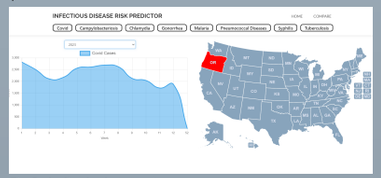

Fig 2 : Individual state view

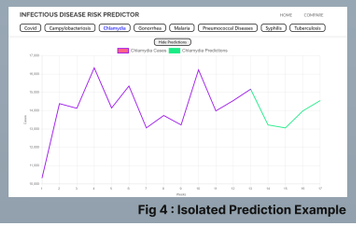

Fig 4 : Isolated Prediction Example

Fig 3 : Comparison View includes state and year weekly data

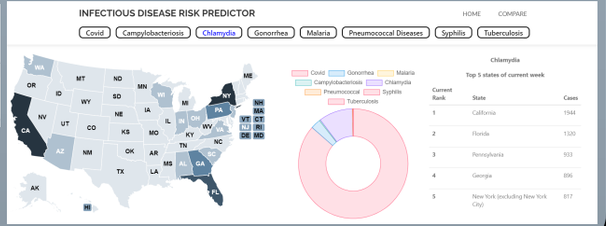

Fig 5 : Homepage of Nation View

Time Series Forecast Analysis

Preparation for Forecasting with Machine Learning
- Data was pre-processed in preparation for analysis including **standardization** and **reshaping.**
- Issues with integrity of data started showing up with daily data as case reporting had dropped off, therefore **resampling** was also needed.
- Once preparation is completed the **feature set** including *population totals, urban population percentage, disease deaths, disease cases, deaths per population, and cases per population* was finalized.

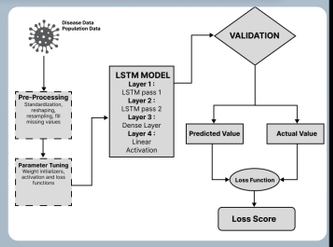

Fig 6 : Forecast Analysis Pipeline

Long Term Short Memory Neural Network
Otherwise known as LSTM Model

- A type of **Recurrent Neural Network** capable of selectively retaining or forgetting information over time.
- The model is based off of the principles of a Neural Network, and uses specialized memory cells to retain or forget to make more accurate predictions.
- **450 plus models** were trained for our specific implementation of LSTM to individually predict each states cases and deaths in 4 week increments.

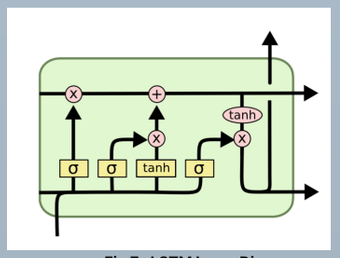

Fig 7 : LSTM Layer Diagram

Experimental Results

LSTM MODEL
- Total data available for training is **91,216 individual data points**, with an average of **1500** individual data points **per model** *with resampling.*
- Our model was then **fine-tuned using our validation set**, which accounts for **20 percent** of our training set.
- Training Size of our models is dependent on the size of data with a **75:25 ratio** of training data to testing data
- When determining the accuracy of our model we used **Root Mean Square Error** as our main performance metric as it compares the difference between the predicted values and the actual values of a dataset. Lower **RSME** values indicate better predictive results.
- After testing we achieved an **average RMSE of .133,** and **an average loss of 3.9 %** across all models.
- When compared to a **Linear Regression model** for the US Covid prediction, we showed lower **RMSE** values using LSTM. Our model returned **.129** while the Regression Model showcased a **.2495**.

COVID PREDICTION METRICS		
LOCALITY	RMSE VALUE	LOSS
OHIO	.433	.188
USA	.129	.016
AVG	.1638	.086

Fig 8 : Sample of Covid Prediction Metrics

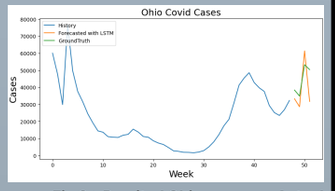

Fig 9 : Precited Ohio cases vs. Actual

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Fig 10 : Formula for Root Mean Square Error

Conclusion and Future Recommendations

Conclusion
We believe that more meaningful analysis of disease risk using **Time Series Forecast Analysis** can show promising changes to our public health. Considering the constraints we ended up facing for our implementation, we feel we reached our initial goal.

Future Recommendations
- Greater access to obtaining both a larger and more concise set of data.
- Improve automation to our application, making it able to both collect all data and run model by itself.
- Adding more data regarding correlating features for our predictive model. (*ie. Infectiousness, Race, Sex, Age*).
- Benchmark against another model (Specifically Vector Auto Regressive Model).