Prediction

This prediction model is a method of time based forecasting based on data collected at multiple points of time using tcpdump. The data that is most likely to be present in the future is predicted and is scrutinised under the detection algorithm in order to determine if the connection is secure or not.

**Input:-**Work datasets(in modified KDD format)

**Output**:-Forecasted datasets(in modified KDD format)

**Step1:-Loading the data as a Time Series**

We’ve selected the time series data type because it is easy to handle the training and forecasting operations on such a data type. To be more specific we have used a **univariate** time series ,meaning such a series contains a variable that depends on only **one** factor ,here **time.**

The tcpdump dataset for **one** ip contains **38 columns** which is split into 38 files,each consisting of :-

a)**Column 1:-** The **time** at which the connection was made, recorded in the following format:-

**hh-mm-ss.ff**

**hh-** hours

**mm-**minutes

**ss-**seconds

**ff-**milliseconds

b)**Column 2:-**The **nth** column values from tcpdump dataset,where (**1<n<38)**

**Step 2:-Making the time series stationary**

A Time Series is said to be stationary if its **statistical properties** such as mean, variance remain **constant over time**. This operation needs to be done because most of the forecasting algorithms work on a stationary series.

There are 2 major factors which make a time series **non-stationary:-**

**a)Trend –** Varying mean over time.

**b)Seasonality –** Variations at specific time frames.

Hence, one method to make a series stationary would be to remove the trend and seasonality components from the series .

It is then possible to apply the statistical forecasting techniques on this series. The final step would be to convert the forecasted values into the original scale by applying trend and seasonality constraints back.

**Eliminating Trend and Seasonality:-**

One of the most common ways to eliminate trend is the **transformation** of the data provided. Due to the erratic nature of our dataset, we have used the **square root** function to rescale all the values to a smaller range. To further remove trend we use a technique named **‘smoothing’.**

 Smoothing refers to taking rolling estimates, i.e. considering the past few instances. We have used ‘**moving averages’** to implement smoothing.

Next we present two methods which equally reduce ‘trend’ and ‘seasonality’:-

**i)Differencing:** Taking the difference with a particular time lag.

In this technique, we take the difference of the observation at a particular instant with that at the previous instant.This is mainly used to remove the temporal dependence. We have used **order-1** differencing(differencing is performed only once).

**ii**) **Decomposition**– modelling both trend and seasonality and removing them from the model.

After isolating ‘trend’ and ‘seasonality’ the **residual** time series becomes **stationary.**

**Step 3:- Forecasting the Time Series**

The statistical model that we have used for forecasting is **SARIMAX**. SARIMAX stands for **Seasonal** **Auto-Regressive Integrated Moving Averages with exogenous factors.** The predictors for the model are (**p,d,q)**

The definitions of these predictors are as follows:-

1. **Number of AR (Auto-Regressive) terms (p):** AR terms are just lags of dependent variable. For instance if p is 5, the predictors for x(t) will be x(t-1)….x(t-5).
2. **Number of MA (Moving Average) terms (q):** MA terms are lagged forecast errors in prediction equation. For instance if q is 5, the predictors for x(t) will be e(t-1)….e(t-5) where e(i) is the difference between the moving average at ith instant and actual value.
3. **Number of Differences (d):** These are the number of nonseasonal differences, i.e. in this case we took the first order difference. So either we can pass that variable and put d=0 or pass the original variable and put d=1. Both will generate same results.

The selection of these parameters was done using **Grid Searching Method.**This method involved selecting a particular range of values for p,d,q. After a particular set of these values were chosen the SARIMAX model was run for one set of values and the predictions were compared against the original values.The **root mean square error(RMSE)** was calculated for each set.The final values for p,d and q were obtained from the set with the least RMSE.

**Final value for p,q,d = (4,1,1)**

-------This ends the selection of the best SARIMAX model.----------

The SARIMAX model then starts the prediction process by analysing the differences between the values in the training set.The model searches for patterns and then forecasts the differences based on these patterns.

The forecasted differences are then added cumulatively to the given values to obtain the forecasted values.

The final step involves **reversing the transformation** process to obtain the values on the original scale.

**Steps 2 and 3 are repeated for each of the 38 columns giving us 38 separate files.**

Each of the 38 files contain only one column. Each column has 10 entries because the number of forecasts that the given program makes is reliable to only 10 entries. After a specific entry the results start repeating.

The final forecasted dataset consists of a 38 x 10 matrix by combining each of the 38 files. It is worthwhile to note that this 38 x 10 matrix is made for **one ip address.**

This dataset forms the **work** dataset for the **detection module**.