M Ű E G Y E T E M 1 7 8 2

Machine learning for intelligent transportation systems

Traffic Prediction (Using Time Series Prediction Methods)

Project Laboratory

Student: Muneeb Ali – ZG0KE2

Instructor: Mohammad Bawaneh

Budapest University of Technology and Economics

2022

Contents

Table OF Figures:

Project Description

Increasing traffic on roads is a major problem globally, causing massive loss to economic productivity. This is primarily due to lack of infrastructure growth compared to the growing number of vehicles on roads due to cost and space constraints. Idea of intelligent transportation systems to reduce road traffic congestion and accidents through advanced safety . Intelligent transportation system (ITS) is the application of sensing, analysis, control, and communications technologies to ground transportation to improve safety, mobility, and efficiency.

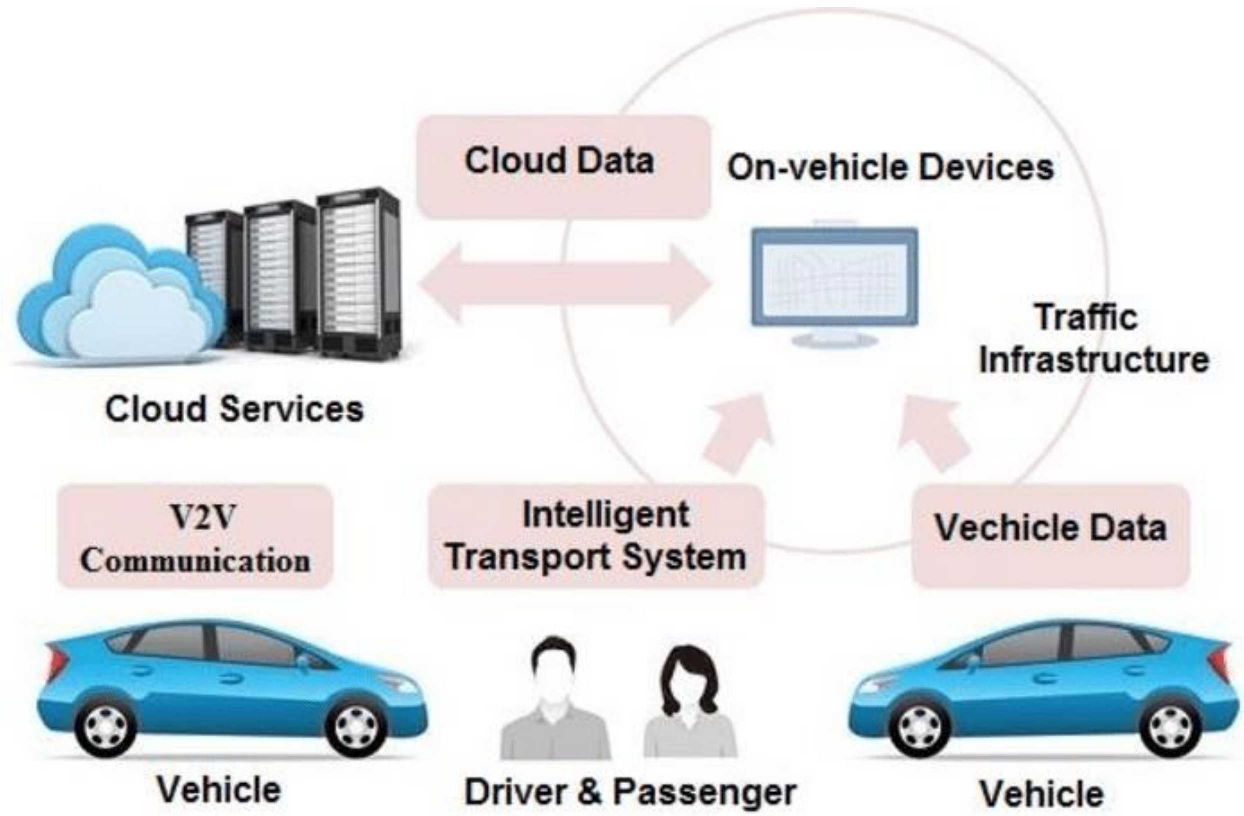
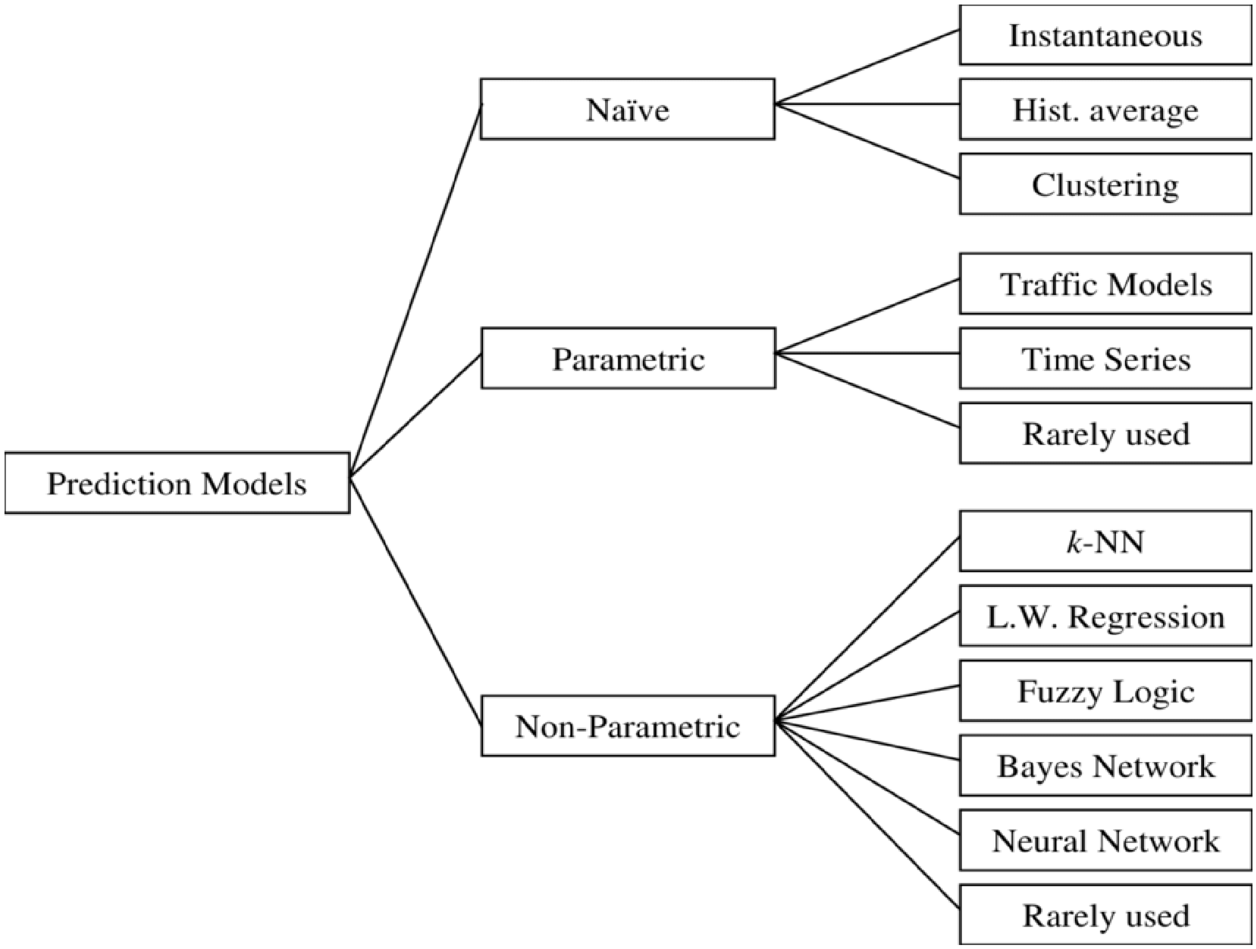


Figure 1: Architectural Evolution of Intelligent Transport Systems (Elzouka, 2016)

In ATMSs and ITSs it is a fundamental challenge to predict the next possible states of traffic with high precision because this information helps to prevent unlikely events like traffic jams or other anomalies on roads (hit.bme.hu, n.d.). Machine learning methods for this purpose will be analyzed and utilized. The main challenge to predict the next possible states of traffic can be solved by using machine learning algorithms like CNN,LSTM and GRU.However, it is hard to select the most appropriate traffic prediction method for one application. In this Project aim to provide different methods for traffic prediction .



[1]*Figure 2: Taxonomy of prediction models (Hinsbergen, 2017)*

Also, the data obtained from sensors need to preprocess . We are getting data from getTraf_data (get MnDOT Traffic Data) is a software utility tool provided to public for retrieving traffic data from the traffic sensors managed by the RTMC (Regional Transportation Management Center), a division of MnDOT (Minnesota Department of Transportation) .This sensor is located at Twin city Minneapolis–Saint Paul (htt1).

In this project focus was on Data Cleaning and Preprocessing and Traffic Prediction (Using Time Series Prediction Methods) using different machine learning algorithms and their efficiencies are calculated and compared .First of all few basic artificial intelligence terms are introduced.

---

[1] This picture shows different possible methods for traffic prediction

Chapter 1 :Basic Artificial Intelligence Terms

Basic statistical terms

Mean (Arithmetic mean): The mean is central value of finite data set.

$$\overline{x} = \frac{x_1 + x_2 + x_{3+\ldots} \ldots + x_n}{n}$$

Other type of means: Geomatic mean, Harmonic mean, For Probability destitution it is expected value $E(x) = \sum x\, P(x)$ where x is random variable and P(x) is probability mass function.

Median: The median is the middle number in a sorted, ascending or descending, list of numbers.

- If there is an odd number in the list, then its middle number is median .
- If there is an even number in the list, then mean of middle numbers determine the median value.

Standard deviation: It shows how much data is averagely scattered around mean value.

Variance: It is a squared value of stand deviation

1st, 2nd, 3rd, and 4th quartile: This divides the number of data points into four parts around medians. Data must be arranged from lowest to highest. It provides information about both the center and the spread of the data. We can calculate mean difference.

Data preprocessing: Data comes from different sources in real world is not real .It can have noisy value, missing value, non-consistence data, non-consistence format, making the date ready and useful is called data preprocessing. In short making data qualitative. We can apply Data cleaning (Missing value, Noise), data transformation and data reduction.

Time series: A time series is a sequence of observation taken at successive equally spaced points in time.

Mean filter: It is a Data-preprocessing non-linear digital filtering technique .It replaces each entry with the mean of neighboring entries. It's mostly used in removing the noise from signal.

Median filter: It replaces each entry with the median of neighboring entries

Exponential smoothing filter: Its use to smooth the time series with exponential window function.

Training set: It is the set of data that is used to train and make the model learn the hidden features/patterns in the data.

Testing set: The test set is a separate set of data used to test the model after completing the training.

Dimensionality reduction

In High-dimensional space are converted into a low-dimensional space. It is used to get Meaningful properties of the original data.

Principal component analysis: It Preserve "useful" information in low dimensional data. In linear dimension reduction PCA is a projection-based method which transforms the data by projecting it onto a set of orthogonal axes.

Algorithm:

- Subtracting the mean of the data from the original dataset
- Finding the covariance matrix of the dataset
- Finding the eigenvector(s) associated with the greatest eigenvalue(s)
- Projecting the original dataset on the eigenvector(s)
- Use only a certain number of the eigenvector(s)
- Do back-projection to the original basis vectors

Classification meaning, and methods

Decision tree:

It is supervised Machine Learning. Here the data is continuously split according to a certain parameter. The leaves are the decisions or the final outcomes

K nearest neighbor:

It is Supervised Learning technique, stores all the available data and classifies a new data point based on the similarities.

Neural network:

It is Deep learning which is Inspired by human brain. Artificial neural networks (ANNs) are comprised of a node layer, input layer, one or more hidden layers, and an output layer.

Regression analysis

It is used to find the relationship between data set point. It has one dependent variable one or more independent variables.

Linear regression Least square method

It minimizes the residue (Observed Value-Fitted model value). We find quadratic Loss Function. For example, the equation of linear Function (in 2D):

Y=m*X+c (c=y-axis intercept of line, m=slop)

ARMA and ARIMA:

ARMA (p, q): This explains the relationship of a time series with both random noise (moving average part) and previous step value (autoregressive part).

ARIMA Model (p, d, q): ARIMA stands for Autoregressive Integrated Moving Average. This model is the combination of autoregression, a moving average model and differencing. Differencing is useful to remove the trend in a time series and make it stationary. The degree of differencing is number of times it was differenced.

Clustering meaning, and methods

It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum.

K-Mean clustering

Algorithm**:**

- Specify number of clusters K.
- Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
- Keep iterating until there is no change to the centroids. i.e., assignment of data points to clusters is not changing.
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).

Hierarchical clustering:

All the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

DBSCAN (Density-based spatial clustering of applications with noise)

It is Density-**b**ased **s**patial **c**lustering of **a**pplications with **n**oise. It used for Arbitrary shaped clusters and clusters with noise. Clusters are dense regions in the data space, separated by regions of the lower density of points. The neighborhood of a given radius must contain at least a minimum number of points.

What is the problem of overfitting, and how to avoid it?

It is due to closely or exactly to a particular set of data. May therefore fail to fit additional data or predict future observations reliably.

Cross Validation: Involves dividing data into a training set and a test set. Fit the model parameters on the training set and evaluate performance on the test set.

k-fold cross-validation:

- Data is first partitioned into k equally (or nearly equally) sized segments or folds.
- Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k - 1 folds are used for learning.

Early stopping: In this method Stop Training when Generalization Error Increases. There are three elements to using early stopping; they are: monitoring model performance, Trigger to stop training and the choice of model to use

Bias-Variance trade off: Bias is the expectation in error and variance is the variability in the model. We must find proper balance between these two parameters.

Anomaly Detection

In Anomaly detection we identify rare events or observations which can raise suspicions by being statistically different from majority of the data.

Modified z- score method:

Z-score is a statistical measure that tells you how far a data point from the rest of the dataset. If that values with modified z-scores less than -3.5 or greater than 3.5 be labeled as potential outliers.

Local Outlier Factor

It's done by measuring the local density deviation of a given data point with respect to the data points near it. Local density is determined by estimating distances between data points that are neighbors (k-nearest neighbors). By comparing these we can check which data points have similar densities and which have a lesser density than its neighbors. The ones with the lesser densities are considered as the outliers.

Chapter 2 :Basic Libraries

NumPy

NumPy stands for Numerical Python. NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. It is an open-source project, and we can use it freely. It Support for large, multi-dimensional arrays and matrices and Scientific computation. It is an Open source. It contains tools for integrating code from C/C++ and Fortran. (Wikipedia, n.d.)

Pandas

It is Python Data Analysis Library. Pandas is an open-source library. It works with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users. It is Fast and efficient for manipulating and analyzing data. Data from different file objects can be loaded. Easy handling of missing data and Columns can be inserted and deleted from Data Frame and higher dimensional objects.

Keras

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. In to standard this project it is used for neural networks, support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.

Sklearn

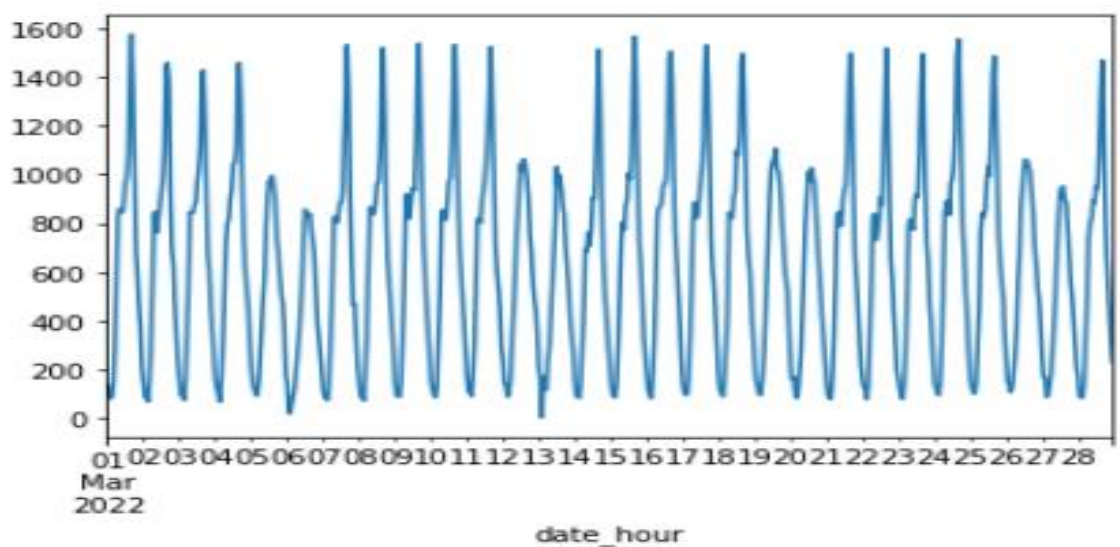
he sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction. In this project it is used for importing some important functions like MinMaxscaling ,square mean error and square log error.

Matplotlib

Most of the Matplotlib utilities lies under the pyplot submodule and are usually imported under the plt for plotting .

Chapter 3 : Data Description

getTraf_data (get MnDOT Traffic Data) is a software utility tool provided to public for retrieving traffic data from the traffic sensors managed by the RTMC (Regional Transportation Management Center), a division of MnDOT (Minnesota Department of Transportation). RTMC manages and collects traffic data in every 30 seconds from the two types of traffic sensors, inductive loop detectors and Wavetronix radar detectors, installed on the Twin Cities' (Minneapolis and St. Paul's) freeway network. The objective of this software was to provide a simple and easy-to-use tool for data retrievals to anybody interested in the RTMC traffic data. Types of data can be retrieved using the getTraf_data tool presently include (1) 30- second volumes, (2) 30-second occupancies, (3) 30-seconds speeds, (4) hourly volumes, (5) daily volumes, (6) hourly average speeds, and (7) hourly speed bins. Additional data types are expected to be added in the future versions. Data that is obtained is for the whole month of March 2022 of sensor 6908.



Figur3:plot of data

Chapter 4 :Preprocessing

Data is retrieved from getTraf_data has missing value and it is also important to transform the data into usable form and format.

Data Cleaning

Missing value is replaced by -1 automatically in the software. Its impotent to fill this value. Forward and backward filling is used to fill the missing value. As we know forward, and backward filling only work on the if there is a missing value. I have replaced -1 by NAN value and then forward and backward filling is applied.

```
HourlyVols['TotalVol']=HourlyVols['TotalVol'].replace(-1,np.nan)
```

```
HourlyVols.ffill()
```

```
HourlyVols.bfill()
```

I can confirm missing value my using this piece of code.

```
print(-1 in HourlyVols['TotalVol'].  unique())
```

False

It is also necessary to format the data. I converted it into datetime to make into data and time one variable. Can date_hour column should be index column.

```
HourlyVols['date_hour'] = pd.to_datetime(HourlyVols['date_hour'], format='%m/%d/%Y %H')
```

It can be visualized .

```
HourlyVols['TotalVol'].plot()
```

Data Smoothing

For smoothing simple exponential smoothing is used. First need to import Simple smoothing function.

```
from statsmodels.tsa.api import SimpleExpSmoothing
```

Then smoothing is done at two different values of exponential smoothing factors.

```
Simple_fit1 = SimpleExpSmoothing(HourlyVols).fit(smoothing_level=0.2,optimized=False)
```

```
Simple_fit2 = SimpleExpSmoothing(HourlyVols).fit(smoothing_level=0.8,optimized=False)
```

Large values mean that the model pays attention mainly to the most recent past observations, whereas smaller values mean more of the history is considered when making a prediction.

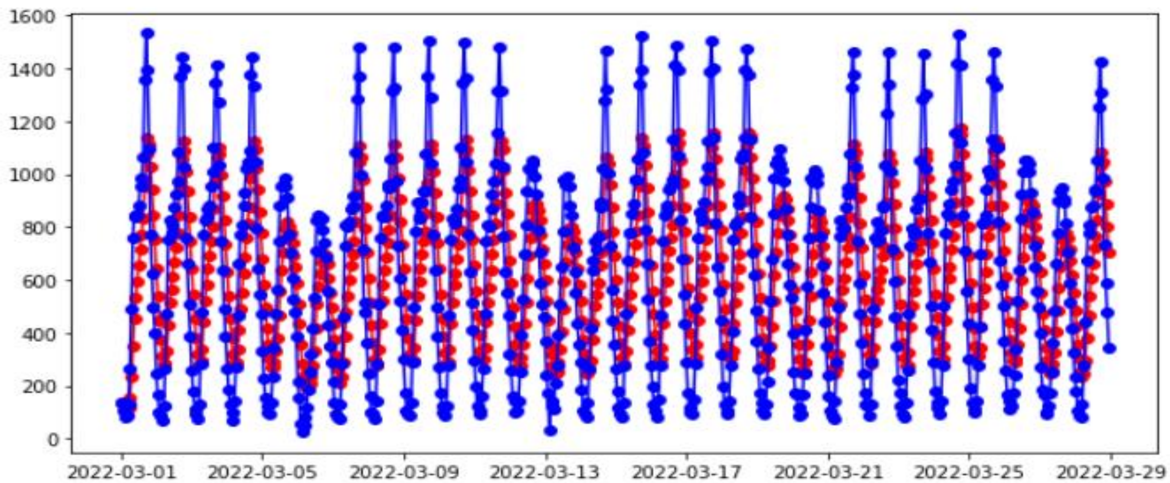


Figure 4:Red shows the curve with smoothing value= 0.2 and blue shows with smoothing value =0.8

Feature scaling

We implement our machine learning model on such datasets. Features with tremendous values dominate those with small values, and the machine learning model treats those with small values as if they do not exist (their influence on the data is not accounted for). To ensure this is not the case, we need to scale our features on the same range, i.e., within the interval of -1 and 1.We also need to caste the Colum into float type

```
df_1=HourlyVols.values
df_1=df_1.astype('float32')
scaler = MinMaxScaler(feature_range=(-1,1))
ts = scaler.fit_transform(df_1)
```

Splitting the dataset into the training and test sets

In machine learning, we split the dataset into a training set and a test set. The training set is the fraction of a dataset that we use to implement the model. On the other hand, the test set is the fraction of the dataset that we use to evaluate the performance the model. The test set is assumed to be unknown during the process of the model implementation. 66 % of data is used for training and 33% is used for testing.

```
timestep = 5
X= []
Y=[]
raw_data=ts
for i in range(len(raw_data)- (timestep)):
  X.append(raw_data[i:i+timestep])
  Y.append(raw_data[i+timestep])
X=np.asanyarray(X)
Y=np.asanyarray(Y)
k = 450
Xtrain = X[:k,:,:]
Ytrain = Y[:k]
```

Chapter 5:Machine Learning Algorithm

CNN

A one-dimensional CNN is a CNN model that has a convolutional hidden layer that operates over a 1D sequence. This is followed by a second convolutional layer in some cases, such as very long input sequences, and then a pooling layer to the output. The convolutional and pooling layers are followed by a dense fully connected layer that interprets the features extracted by the convolutional part of the model. A flatten layer is used between the convolutional layers and the dense layer to reduce the feature maps to a single one-dimensional vector. We can define a 1D CNN Model for univariate time series forecasting as follows:

```
model = Sequential()
model.add(Conv1D(filters=128, kernel_size=2, activation='relu',
input_shape=(5, 1)))
model.add(Conv1D(filters=128, kernel_size=2, activation='relu'))
model.add(Conv1D(filters=128, kernel_size=2, activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
# fit model
model.fit(Xtrain, Ytrain, epochs=200, verbose=0)
```

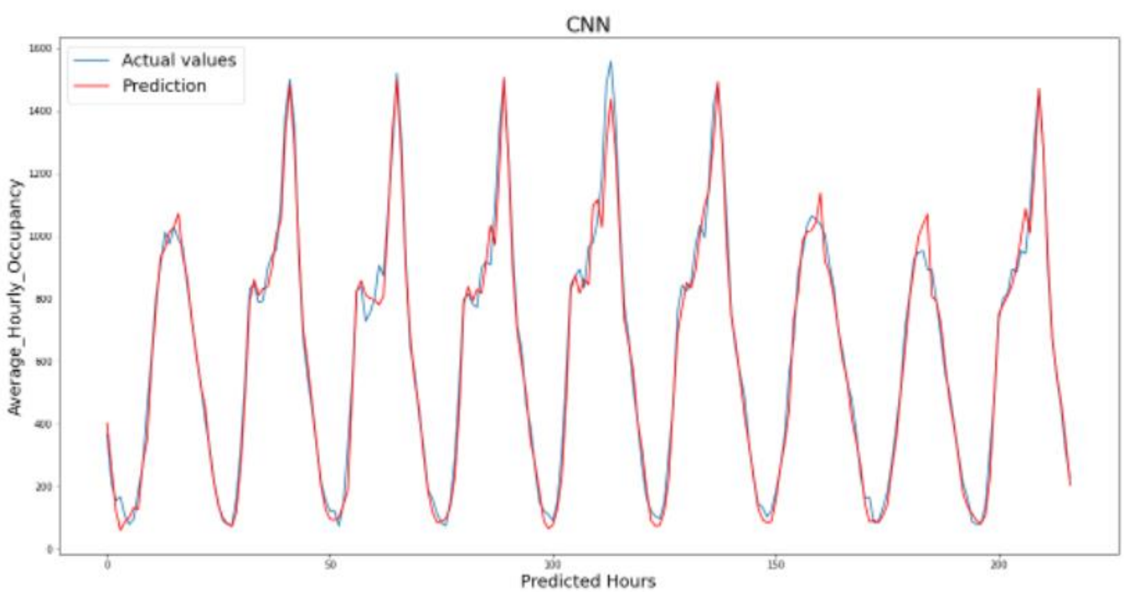


Figure 5:Plot of actual value and predicted value from CNN model

mean_squared_error(Ytest,preds)

2813.459

It has percentage accuracy of 97.571 %.

We can also observe the square error of the
predicted values and actual values which gives
the continues plot of square error at every point

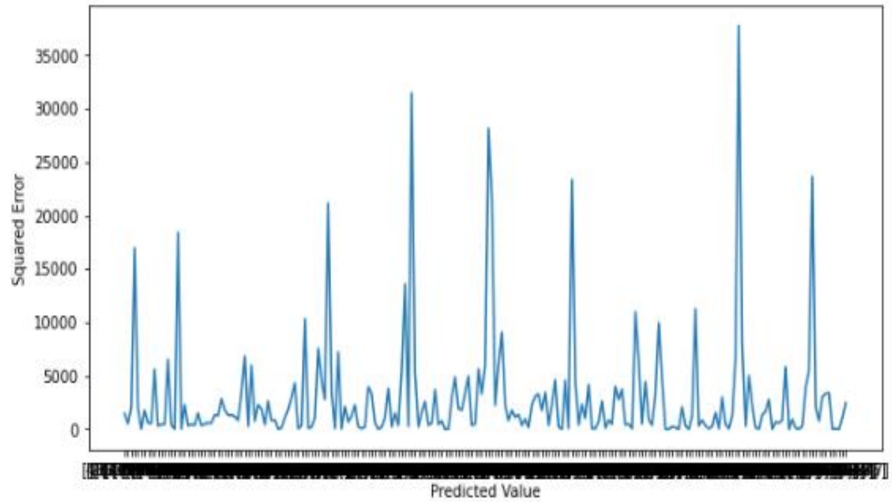


Figure 6:Line Plot of the Increase Square Error with Predictions in CNN

LSTM

LSTM model sees the input data as a sequence, so it can learn patterns from sequenced data better s, especially patterns from long sequences. Long short-term memory (LSTM) units (or blocks) are a building unit for layers of a recurrent neural network (RNN). A RNN composed of LSTM units is often called an LSTM network.

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM. Each of the three gates can be thought of as a "conventional" artificial neuron, as in a multi-layer (or feedforward) neural network: that is, they compute an activation (using an activation function) of a weighted sum. Intuitively, they can be thought as regulators of the flow of values that goes through the connections of the LSTM; hence the denotation "gate". There are connections between these gates and the cell.

The expression long short-term refers to the fact that LSTM is a model for the short-term memory which can last for a long period of time. An LSTM is well-suited to classify, process and predict time series given time lags of unknown size and duration between important events. LSTMs were developed to deal with the exploding and vanishing gradient problem when training traditional RNNs.

```
model_lstm = Sequential()
model_lstm.add(LSTM(50, activation='relu', input_shape=(5,1)))
model_lstm.add(Dense(1))
model_lstm.compile(loss='mse', optimizer='adam')
model_lstm.fit(Xtrain, Ytrain, epochs=200, verbose=0)
model_lstm.summary()
```

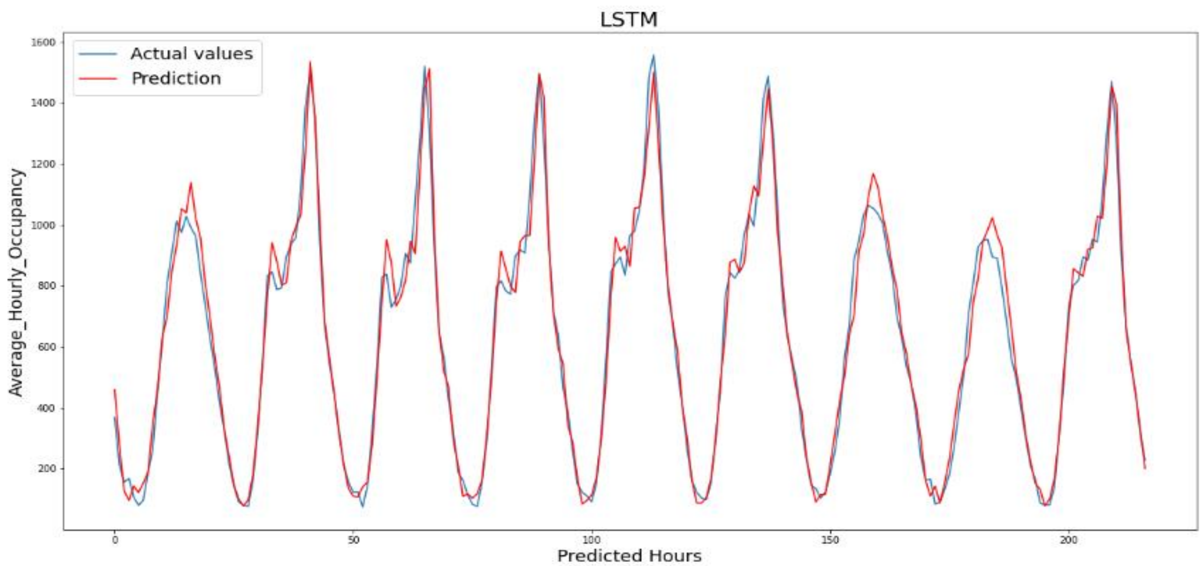


Figure 7:Plot of actual value and predicted value from LSTM model

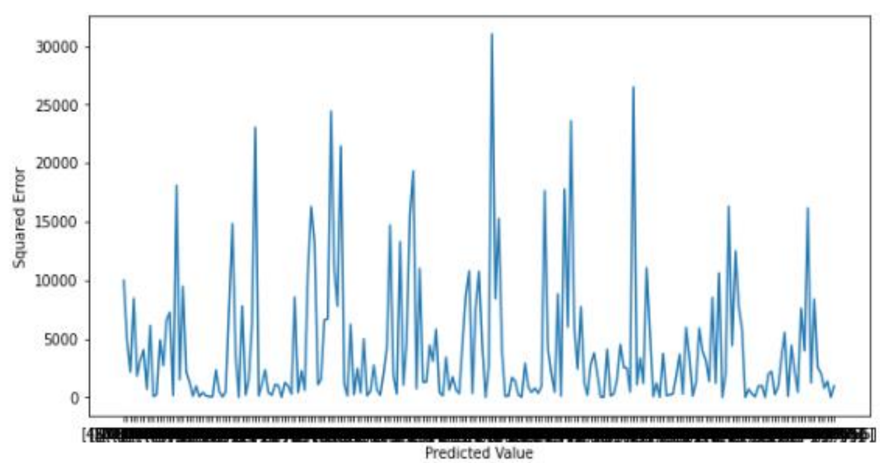
mean_squared_error(Ytest,preds2)

```
4859.497
```

It has percentage accuracy of 96.9 %.

We can also observe the square error of the predicted values and actual value.

Figure 8 Line Plot of the Increase Square Error with Predictions in LSTM



GRU

GRUs are easier to train than LSTMs .In simple words, the GRU unit does not have to use a memory unit to control the flow of information like the LSTM unit. It can directly make use of all hidden states without any control. GRUs have fewer parameters and thus may train a bit faster or need less data to generalize. But, with large data, the LSTMs with higher expressiveness may lead to better results.

They are almost like LSTMs except that they have two gates: reset gate and update gate. Reset gate determines how to combine new input to previous memory and update gate determines how much of the previous state to keep. Update gate in GRU is what input gate and forget gate were in LSTM. We do not have the second nonlinearity in GRU before calculating the output, neither they have the output gate.

```
regressorGRU = Sequential()

regressorGRU.add(GRU(50, activation='relu', input_shape=(5,1)))

regressorGRU.add(Dense(1))

regressorGRU.compile(loss='mse', optimizer='adam')

regressorGRU.fit(Xtrain, Ytrain, epochs=200, verbose=0)

regressorGRU.summary()
```

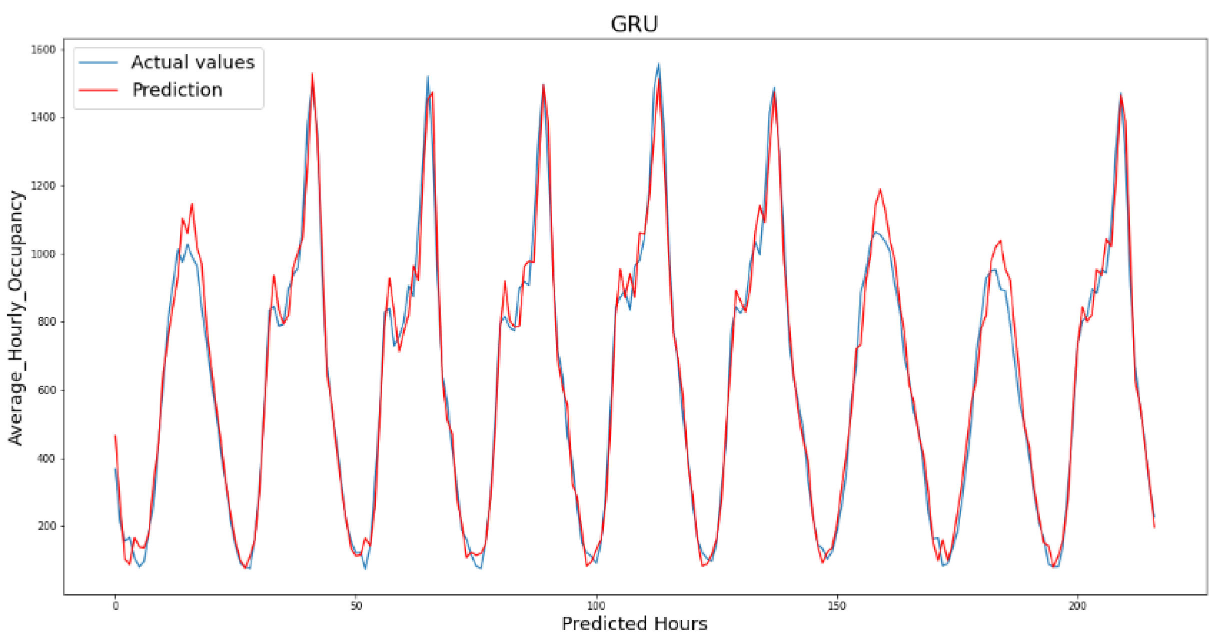


Figure 9 Plot of actual value and predicted value from GRUmodel

mean_squared_error(Ytest,preds3)

3679.8813

`It has percentage accuracy of` 97.2 **%.**

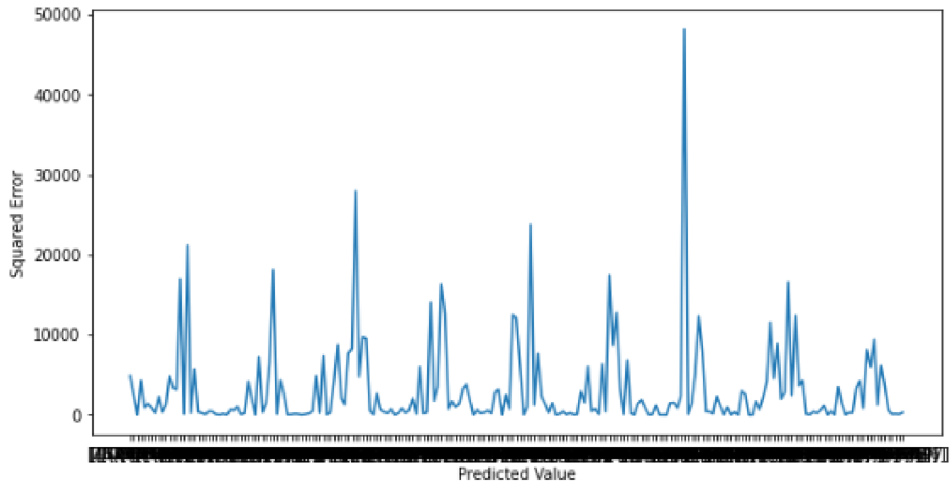


Figure 10:Line Plot of the Increase Square Error with Predictions in GRU

Conclusion

We can see compare the results from these three methods and compare them for given Time series data .CNN is a method that is used to find out the spatial correction of data set. Which is more efficient in our case.Other two methods Long short-term memory (LSTM) and Gated recurrent units are the examples of recurrent neural network (RNN). They are used to solve the problem of Gradient decent in RNN model.

A reset gate and an update gate are used by the GRU. To forget the previous state, the reset gate sits between the previous activation and the next activation, and the update gate decides how much of the activation to use in updating the cell state.

LSTMs manage the exposure of memory content (cell state), whereas GRUs expose a whole cell to other network units. The LSTM has separate input and forget gates, whereas the GRU performs both means for providing through its reset gate.

GRUs is simpler ,faster to train and perform better LSTM for given Time series data .

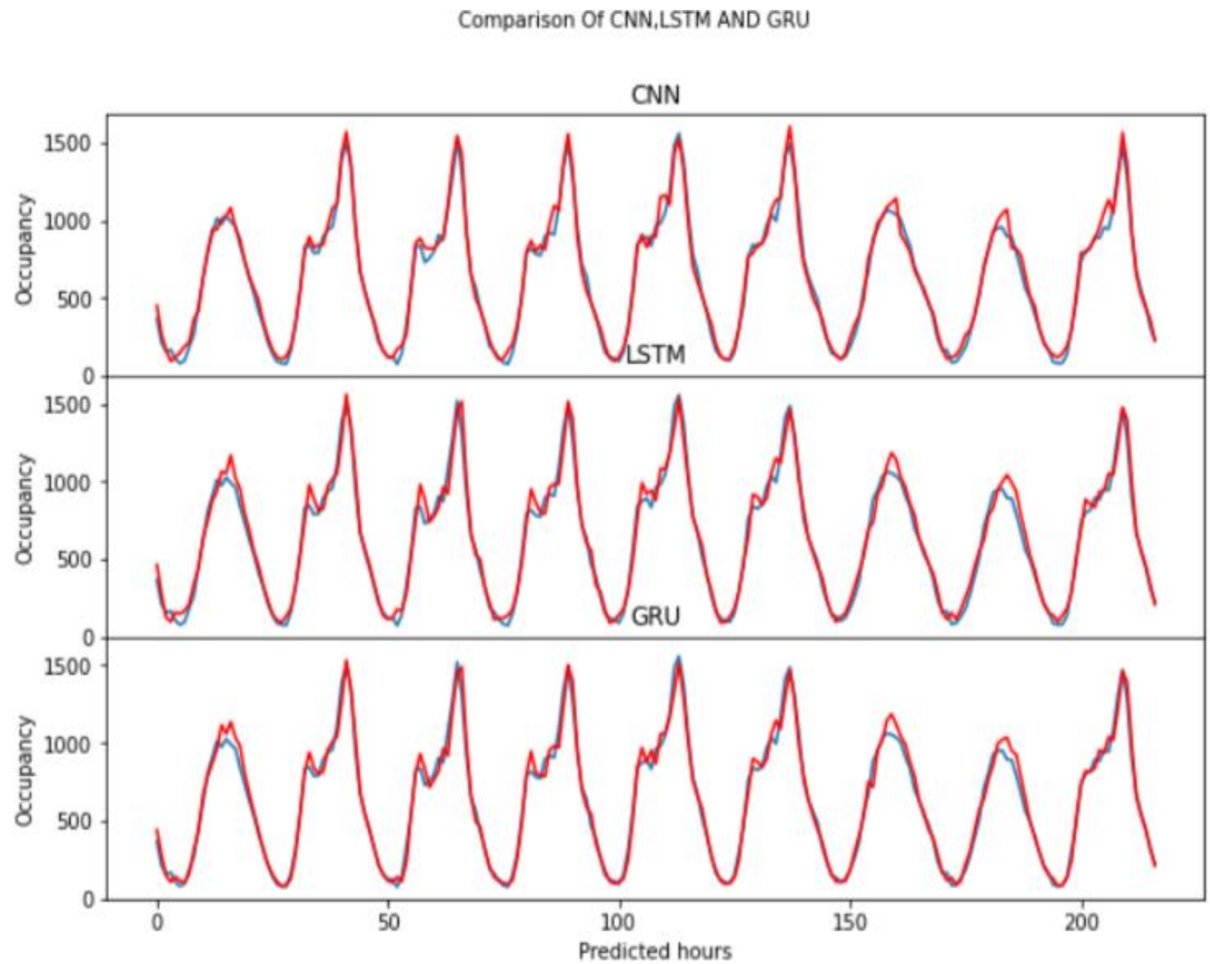


Figure 11Plot of actual value and predicted value three model

There is compare the Mean square errors of these three models .CNN more valid and have less error as compared to the LSTM and GRU. So, for the production modelling for this Time series forecasting CNN is best choice .It is well predicting the patterns and trends than other two experimented models.

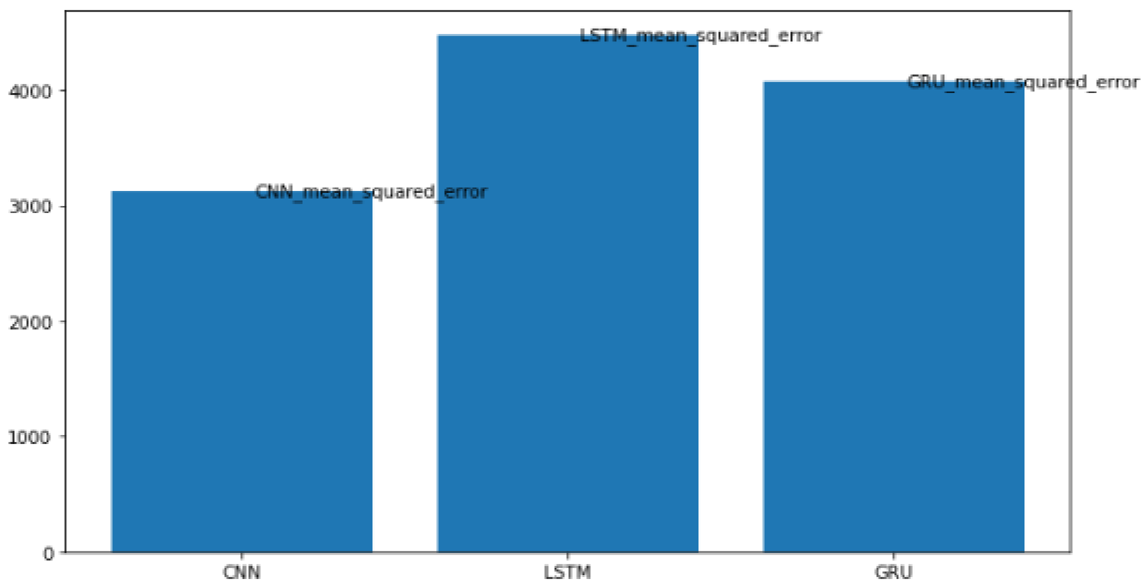
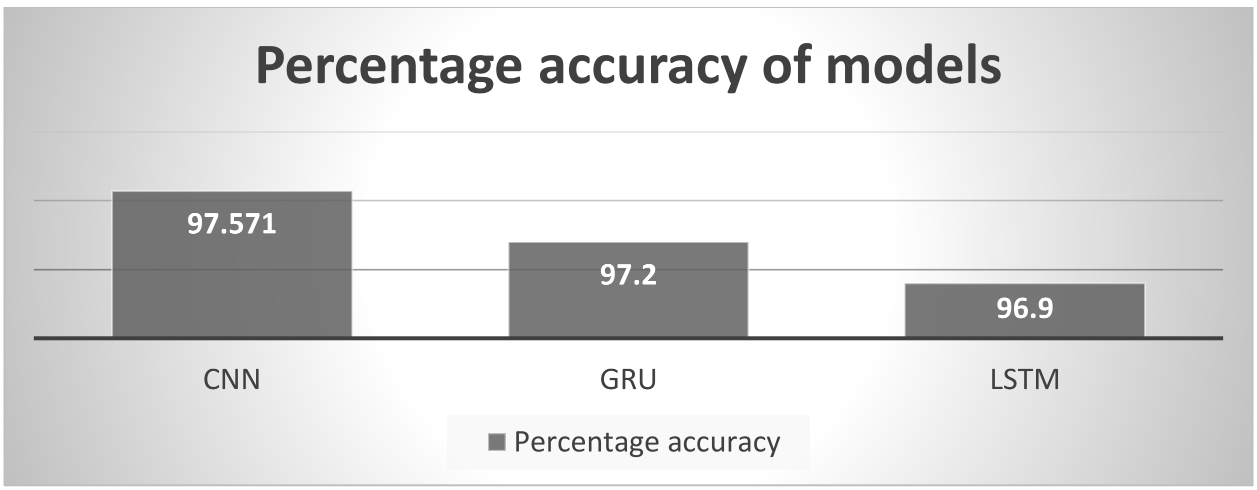


Figure 12Mean squired error of three models



This table compare these three models.

Properties	CNN	LSTM	GRU
Percentage accuracy	97.571	96.9	97.2
Mean Squared Error	2733.205	4721.037	4678.801
Root Mean Squared Error	52.280067	68.7098	68.40176
Mean Absolute Error	39.30431	55.280125	55.81744
Mean Squared Log Error	0.025852023	0.033091076	0.034688294

Works Cited

(n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/NumPy

(n.d.). Retrieved from https://www.cts.umn.edu/news/2021/november/software

Elzouka, H. (2016). A Secure Interactive Architecture for Vehicular Cloud Environment. *Proceedings of the IEEE*.

Hinsbergen, C. L. (2017). Short Term Traffic Prediction Models. *14th World Congress on Intelligent Transport Systems, ITS 2007*.

hit.bme.hu. (n.d.). Retrieved from https://www.hit.bme.hu/edu/project/data?id=19840