A deep learning approach for analysis of IOT data

By

Chayan Maity
Tamalika Halder
Alolika Chakraborty
Dipayita Basu

UNDER THE GUIDANCE OF

Mrs. Sukla Banerjee

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF INFORMATION TECHNOLOGY AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Session 2017-2018

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY [Affiliated to West Bengal University of Technology] CANAL SOUTH ROAD, BELIAGHATA, KOLKATA-700105

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY

TO WHOM IT MAY CONCERN

I hereby recommend that the Project entitled **A deep learning approach for analysis of IOT data** prepared under my supervision by

Chayan Maity (Reg. No 141170110025**, Class Roll No**. CSE/2014/079**)**
Tamalika Halder (Reg. No 141170110087**, Class Roll No**. CSE/2014/078**)**
Alolika Chakraborty (Reg. No 141170110003**, Class Roll No**. CSE/2014/095**)**
Dipayita Basu(Reg. No 141170110030**, Class Roll No**. CSE/2014/089**)**

of B.Tech 8th Semester, may be accepted in partial fulfillment for the degree of **Bachelor of Technology in Computer Science & Engineering** under **Maulana Abdul Kalam Azad University of Technology** (**MAKAUT**).

…….....

Project Supervisor

Department of Computer Science and Engineering

RCC Institute of Information Technology

Countersigned:

…….....

Head

Department OF Computer Sc. & Engg,

RCC Institute of Information Technology

Kolkata- 700105

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY

CERTIFICATE OF APPROVAL

The foregoing Project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

FINAL EXAMINATION FOR

EVALUATION OF PROJECT

1. .....

2. .....

(Signature of Examiners)

ACKNOWLEDGEMENT

I take the opportunity to express my profound gratitude and deep regards to our mentor Mrs. Sukla Banerjee for her exemplary guidance, monitoring and con-stant encouragement throughout the course of this project. Her relentless effort for teaching in the right way and in the correct manner has helped us to attain a high standard in this aspect.

Also, not to forget the coordinated work of our group and hereby we thank each other for the successful completion of the documentation.

.....

Chayan Maity (CSE/2014/079)
Tamalika Halder(CSE/2014/078)
Alolika Chakraborty(CSE/2014/095)
Dipayita Basu(CSE/2014/089)

Table of Contents

Introduction

Internet of things refers to network of objects, each of which has an unique IP address & can connect to internet. These objects can be people, animal and day to day devices like your refrigerator and your coffee machine. These objects can connect to internet (and to each other) and communicate with each other through this net, in ways which have not been thought before. These devices would interact and communicate – to humans or to other machines, as appropriate. These would have embedded controllers to switch things on and off.
Since IoT will be among the greatest sources of new data, data science will make a great contribution to make IoT applications more intelligent. Data science is the combination of different fields of sciences that uses data mining, machine learning and other techniques to find patterns and new insights from data.
The process of applying data analytics methods to particular areas involves defining data types such as volume, variety, velocity; data models such as neural networks, classification, clustering methods and applying efficient algorithms that match with the data characteristics.

IoT is a combination of embedded technologies regarding wired and wireless communications, sensor and actuator devices, and the physical objects connected to the Internet . One of the long-standing objectives of computing is to simplify and enrich human activities and experiences.
IoT needs data to either represent better services to users or enhance IoT framework performance to accomplish this intelligently. In this manner, systems should be able to access raw data from different resources over the network and analyze this information to extract knowledge.

Review of Literature

A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data by C. Narendra Babu, B. Eswara Reddy

A suitable combination of linear and nonlinear models provides a more accurate prediction model than an
individual linear or nonlinear model for forecasting time series data originating from various applications.
The linear autoregressive integrated moving average (ARIMA) and nonlinear artificial neural network
(ANN) models are explored in this paper to devise a new hybrid ARIMA–ANN model for the prediction of time series data. Many of the hybrid ARIMA–ANN models which exist in the literature apply an ARIMA model to given time series data, consider the error between the original and the ARIMA-predicted data as a nonlinear component, and model it using an ANN in different ways. Though these models give predictions with higher accuracy than the individual models, there is scope for further improvement in the accuracy if the nature of the given time series is taken into account before applying the models. In the work described in this paper, the nature of volatility was explored using a moving-average filter, and then an ARIMA and an ANN model were suitably applied. Using a simulated data set and experimental data sets such as sunspot data, electricity price data, and stock market data, the proposed hybrid ARIMA–ANN model was applied along with individual ARIMA and ANN models and some existing hybrid ARIMA–ANN models. The results obtained from all of these data sets show that for both one-step-ahead and multistep-ahead forecasts, the proposed hybrid model has higher prediction accuracy.

A Four-Stage Hybrid Model for Hydrological Time Series Forecasting by Chongli Di, Xiaohua Yang, Xiaochao Wang

Hydrological time series forecasting remains a difficult task due to its complicated nonlinear, non-stationary and multi-scale characteristics. To solve this difficulty and improve the prediction accuracy, a novel four-stage hybrid model is proposed for hydrological time series forecasting based on the principle of 'denoising, decomposition and ensemble'. The proposed model has four stages, i.e., denoising, decomposition, components prediction and ensemble. In the denoising stage, the empirical mode decomposition (EMD) method is utilized to reduce the noises in the hydrological time series. Then, an improved method of EMD, the ensemble empirical mode decomposition (EEMD), is applied to decompose the denoised series into a number of intrinsic mode function (IMF) components and one residual component. Next, the radial basis function neural network (RBFNN) is adopted to predict the trend of all of the components obtained in the decomposition stage. In the final ensemble prediction stage, the forecasting results of all of the IMF and residual components obtained in the third stage are combined to generate the final prediction results, using a linear neural network (LNN) model. For illustration and verification, six hydrological cases with different characteristics are used to test the effectiveness of the

proposed model. The proposed hybrid model performs better than conventional single models, the hybrid models without de-noising or decomposition and the hybrid models based on other methods, such as the wavelet analysis (WA)-based hybrid models. In addition, the denoising and decomposition strategies decrease the complexity of the series and reduce the difficulties of the forecasting. With its effective denoising and accurate decomposition ability, high prediction precision and wide applicability, the new model is very promising for complex time series forecasting. This new forecast model is an extension of nonlinear prediction models.

Hybridization Model of Linear and Nonlinear Time Series Data for Forecasting by Roselina Sallehuddin, Siti Mariyam Shamsuddin, Siti Zaiton Mohd Hashim

The aim of this paper is to propose a novel approach in hybridizing linear and nonlinear model by incorporating several new features. The intended features are multivariate information, hybridization succession alteration, and cooperative feature selection. To assess the performance of the proposed hybrid model allegedly known as Grey Relational Artificial Neural Network(GRANN_ARIMA), extensive
comparisons are done with individual model (Artificial Neural Network(ANN), Autoregressive integrated Moving Average(ARIMA) and Multiple Linear Regression(MR)) and conventional hybrid model (ARIMA_ANN) with Root Mean Square Error(RMSE), Mean Absolute Deviation(MAD), Mean Absolute Percentage Error (MAPE) and Mean Square error(MSE). The experiments have shown that the proposed hybrid model has outperformed other models with 99.5% forecasting accuracy for small-scale data and 99.84% for large-scale data. The obtained empirical results have also proved that the GRANN-ARIMA is more accurate and robust due to its promising performance and capability in handling small and large scale time series data. In addition, the implementation of cooperative feature selection has assisted the forecaster to automatically determine the optimum number of input factor amid with its important ness and consequence on the generated output.

R2N2: Residual Recurrent Neural Networks
for Multivariate Time Series Forecasting by Hardik Goel , Igor Melnyk and Arindam Bannerjee

Multivariate time-series modeling and forecasting is an important problem with numerous applications.
Traditional approaches such as VAR (vector auto-regressive) models and more recent approaches such as RNNs (recurrent neural networks) are indispensable tools in modeling time-series data. In many multivariate time series modeling problems, there is usually a significant linear dependency component,for which VARs are suitable, and a nonlinear component, for which RNNs are suitable. Modeling such times series with only VAR or only RNNs can lead to poor predictive performance or complex models with large training times. In this work, we propose a hybrid model called R2N2 (Residual RNN), which first models the time series with a simple linear model (like VAR) and then models its residual errors using RNNs. R2N2s can be trained using existing algorithms for VARs and RNNs. Through an extensive empirical evaluation on two real world datasets (aviation and climate domains), we show that R2N2 is competitive, usually better than VAR or RNN, used alone. We also show that R2N2 is faster to train as compared to an RNN, while requiring less number of hidden units.

Objective of the Project

Our primary goal is to analyze different features of a smart city such as Air Pollution Control

At first we are focusing on Air Quality Control. Here we have taken a dataset of Denmark, where air quality index of Different gases like So2, No2, Ozone, Co2, particulate matter are stored. Our key con-cern is to analyze these data and generate predictive modeling to prevent or take safety measure before the air quality of that location detoriates and becomes hazardous.

We have proposed to control traffic according to pollution in the smart city. From the dataset we have acquired we are trying to forecast the air quality indexes and generate an alarm with unhealthy conditions at various intersections. This will result in optimized air pollution and benefit the society.

Data Science:

Data science is the study of where information comes from, what it represents and how it can be turned into a valuable resource in the creation of business and IT strategies. Mining large amounts of structured and unstructured data to identify patterns can help an organization rein in costs, increase efficiencies, recognize new market opportunities and increase the organization's competitive advantage.

Application:

The data science field employs mathematics, statistics and computer science disciplines, and incorporates techniques like machine learning, cluster analysis, data mining and visualization.

Time-series data:

Time series is simply a set of observations recorded at different time points. Since time runs forward, time series observations has a natural ordering.
Time series analysis is a statistical technique that deals with time series data, or trend analysis. Time series data means that data is in a series of particular time periods or intervals.

Multivariate data:

Multivariate data is the data in which analysis are based on more than two variables per observation. Usually multivariate data is used for explanatory purposes.

Multivariate Data Analysis refers to any statistical technique used to analyze data that arises from more than one variable. This essentially models reality where each situation, product, or decision

involves more than a single variable. The information age has resulted in masses of data in every field. Despite the quantum of data available, the ability to obtain a clear picture of what is going on and make intelligent decisions is a challenge. When available information is stored in database tables containing rows and columns, Multivariate Analysis can be used to process the information in a meaningful fashion.

Disadvantages Data Analytics:

Following are the disadvantages of Data Analytics:

➡This may breach privacy of the customers as their information such as purchases, online transactions, subscriptions are visible to their parent companies.

➡The cost of data analytics tools vary based on applications and features supported. Moreover some of the data analytics tools are complex to use and require training.

➡The information obtained using data analytics can also be misused against group of people of certain country or community or caste.

➡It is very difficult to select the right data analytics tools. This is due to the fact that it requires knowledge of the tools and their accuracy in analyzing the relevant data as per applications. This increases time and cost to the company.

How we can use deep learning for analyzing time series and multivariate data?

Deep learning can capture dependencies and patterns so complex, that no other algorithm is able to recognize. Of course, it comes with a huge computational cost and some other engineering complexities, but it allows us to solve new data mining tasks which were poorly solved before such as Complex time series analysis.

Deep learning can be used for image recognition, object detection, text processing tasks .For example Recurrent Neural Network (RNN) using the Long Short Term Memory (LSTM) is used for sentiment analysis.

System Design

System Requirements:

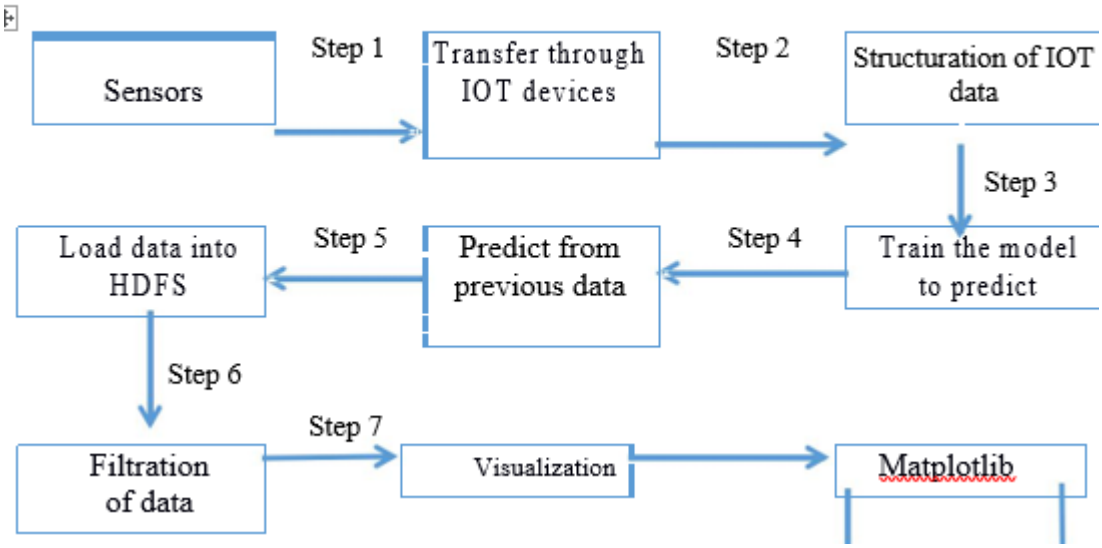
Hardware Requirement:

- 64-bit CPU that supports 64bit virtualization - and of course 64-bit OS is needed
- At least 8GB RAM required (16GB recommended)

Software Requirement:

- Linux installed.
- Anaconda installed.
- Spyder installed.
- There should have updated version of Python (at least 3.0). Li-braries like Pandas, Keras, Numpy, Tensorflow, Scikit learn is required.

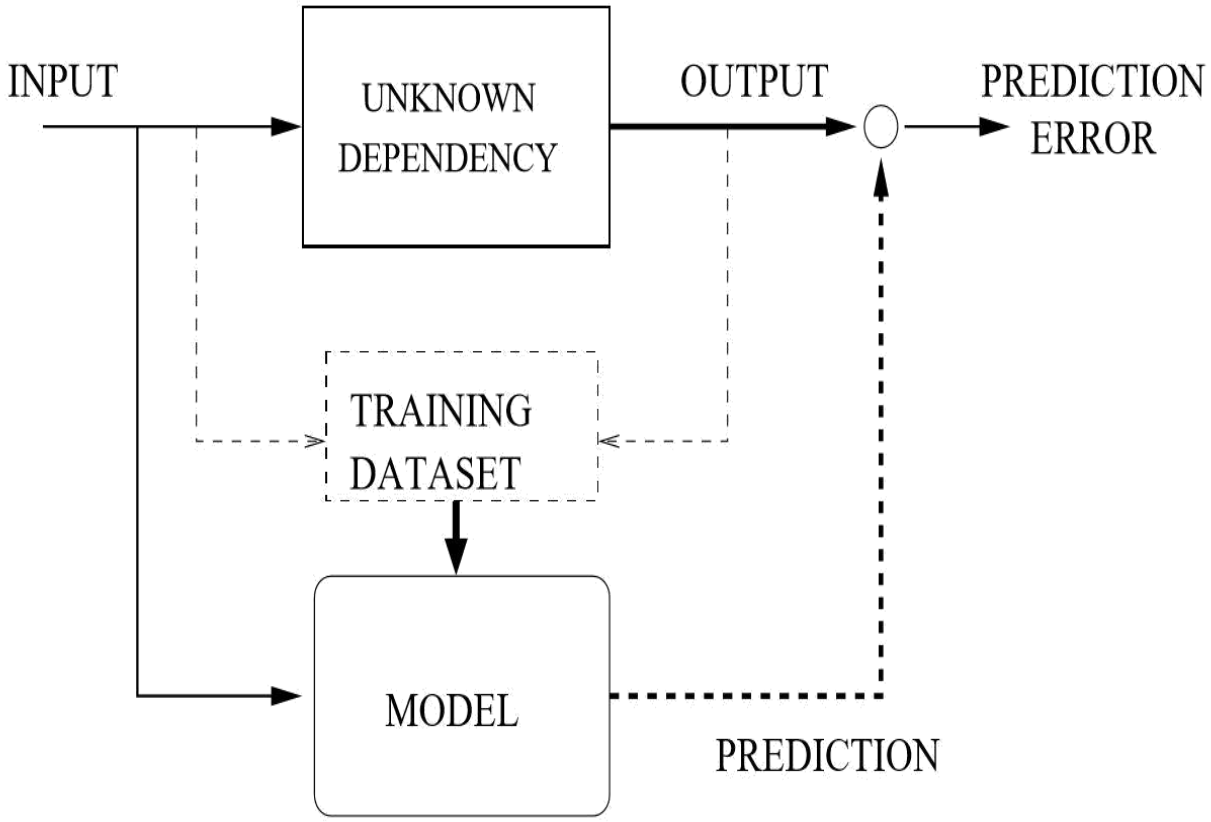
System Architecture:



Methodology for Implementation (Formulation & Algorithm)

Machine Learning Algorithms & Time Series Analysis: Machine learning is a subset of artificial intelli-gence that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs, those can teach themselves to grow and change when exposed to new data.

We consider the prediction problem as a problem of supervised learning problem, where we have to infer from historical data the possibly nonlinear dependence between the input (past embedding vector) and the output (future value).

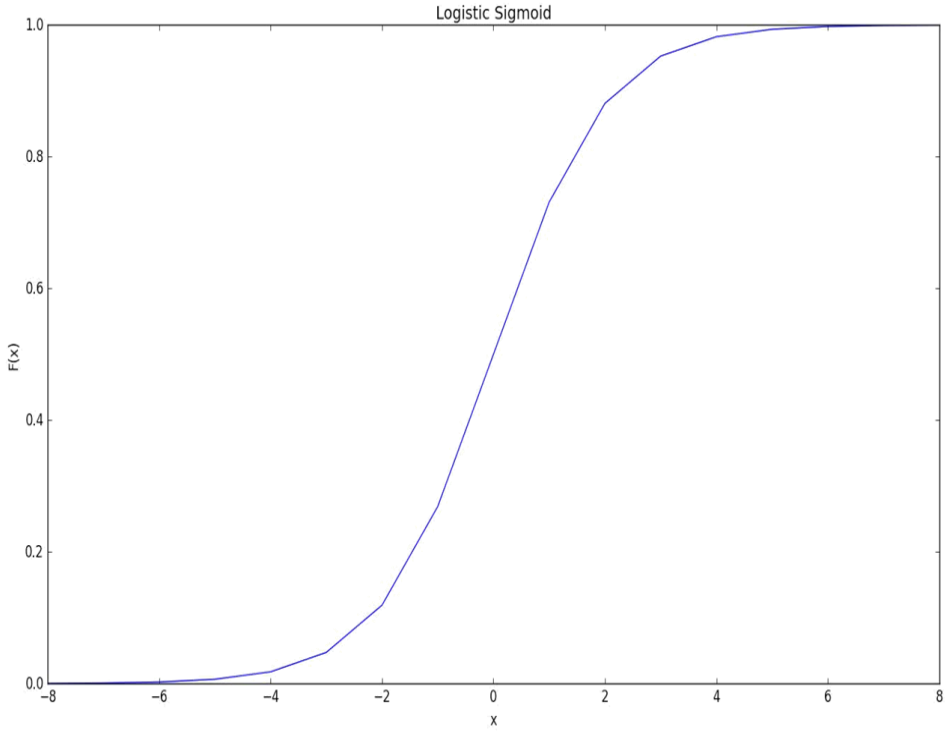


Neural Network

For this prediction task we are using *Backpropagation Neural Network*. A neural Network is a machine learning algorithm to perform *Classification* and *Regression* related task. Here we are using LSTM to predict the temperature by first training the network with 4000 observations. While training we are using Dew point, Humidity, Wind Direction, Wind Speed as features. We are using *logistic sigmoid* as activa-tion function for the neural network. A logistic sigmoid function always outputs between 0 and 1 (both inclusive).

$$F(x) = 1 / (1 + exp(-x))$$

The derivative of sigmoid function $F'(x) = F(x)(1 – F(x)$



From the above graph it is clear that sigmoid function's activation range is between –4 to +4. Outside this range the output is almost 0 or 1. A typical neural network has three layers. One input layer, one output layer and one hidden layer. The number of nodes in the input layer is euqal to the number of feature in dataset. For number of nodes in hidden layer there is no specific rule. The number of node in output layer is equal to number of feature in output set. The nodes are connected through weighted edges. The weights are at first initialized randomly.

FeedForward:- The inputs from the input layer to the hidden layer are multiplied with the respective weights and then the each hidden node sums up all the inputs it is getting. Then the value is passed through the activation function and again the values from hidden layer to output layer are multiplied by respective weights and the output sums up the input it is receiving, then it passes the sum through the activation again and produces output.

Backpropagation:- The output from the output layer is then compared with the target output. Our goal with backpropagation is to update each of the weights in the network so that they cause the actual output to be closer the target output, thereby minimizing the error for each output neuron and the network as a whole.

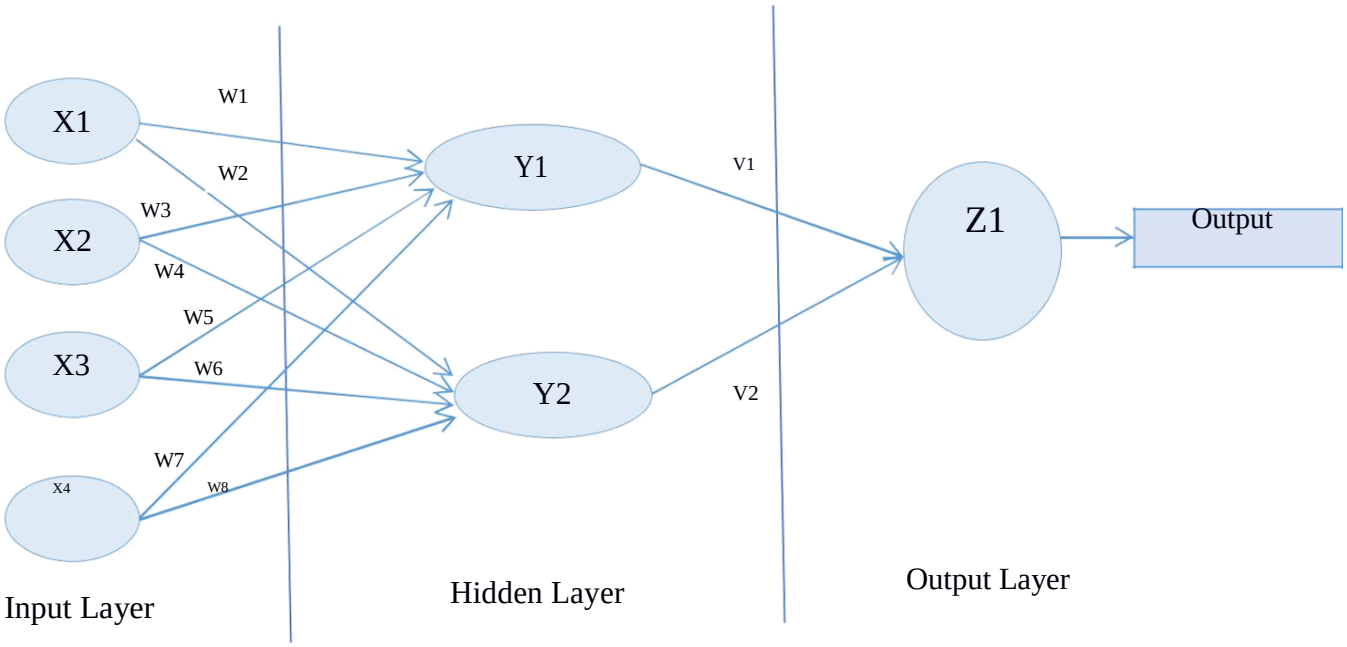


Fig. Neural Network(different layers)

Training Algorithm:-

Step 0. Initialize Weights to small random numbers

Feedforward

Step 1. Until RMS Error is very small do

Step 2. Y1 = x1 * w1 + x2 * w3 + x3 * w5 + x4 * w7

Y2 = x1 * w2 + x2 * w4 + x3 * w6 + x4 * w8

Step 3. Y1_sigmoid = F(Y1)

Y2_sigmoid = F(Y2)

Step 4. Z1 = v1 * Y1_sigmoid + v2 * Y2_Sigmoid

Z1_sigmoid = F(Z1)

Backpropagation

Step 5. error = (target – Z1_sigmoid)

Delta = error * F'(Z1)

Delta is error information term

Delta_V1 = Delta * Y1_sigmoid

Delta_V2 = Delta * Y2_sigmoid

Delta_V1 and Delta_V2 are weight correction

terms for V1 and V2 weight

Step 6.

Calculate Weight Updation Terms

Delta_2 = Delta * V1 + Delta * V2

Delta_2_1 = Delta_2 * F'(Y1)

Delta_2_2 = Delta_2 * F'(Y2)

Delta_W1 = Delta_2_1 * X1, Delta_W3 = Delta_2_1 * X2,

Delta_W5 = Delta_2_1 * X3, Delta_W7 = Delta_2_1 * X4

Delta_W2 = Delta_2_2 * X1, Delta_W4 = Delta_2_2 * X2

Delta_W6 = Delta_2_2 * X3, Delta_W8 = Delta_2_2 * X4

Step 7.

Weight Updation

W1 = W1 + Delta_W1, W2 = W2 + Delta_W2

W3 = W3 + Delta_W3, W4 = W4 + Delta_W4

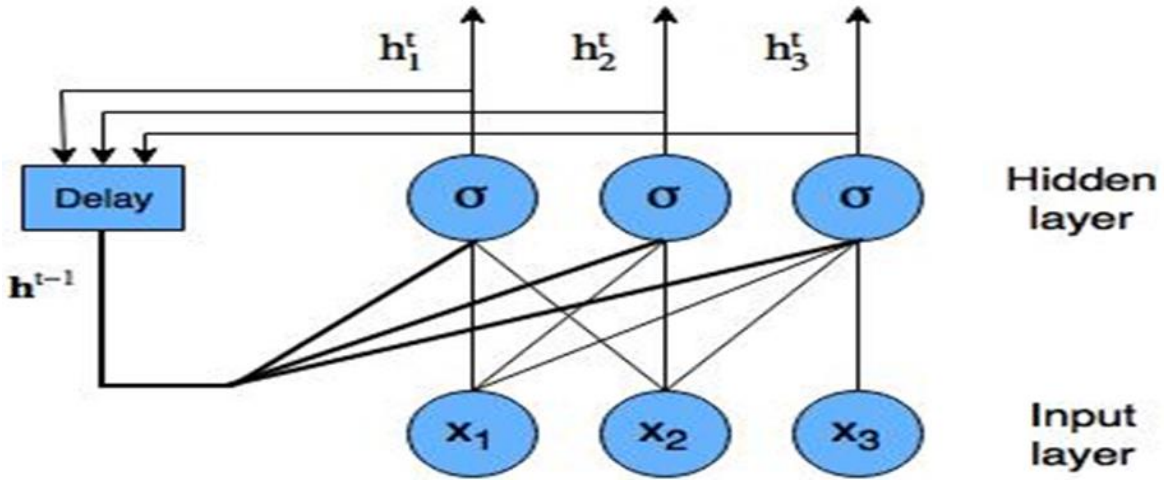
W5 = W5 + Delta_W5, W6 = W6 + Delta_W6

W7 = W7 + Delta_W7, W8 = W8 + Delta_W8

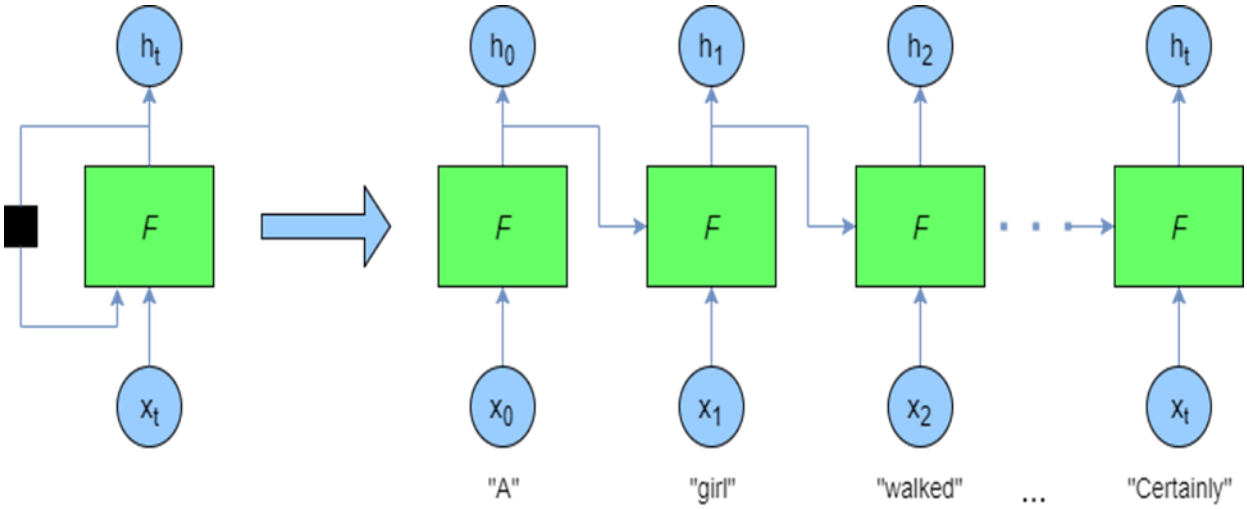
V1 = V1 + Delta_V1, V2 = V2 + Delta_V2

Recurrent neural networks:

A LSTM network is a kind of recurrent neural network. A recurrent neural network is a neural network that attempts to model time or sequence dependent behavior – such as language, stock prices, and electricity demand and so on. This is performed by feeding back the output of a neural network layer at time t to the input of the same network layer at time t + 1. It looks like this:

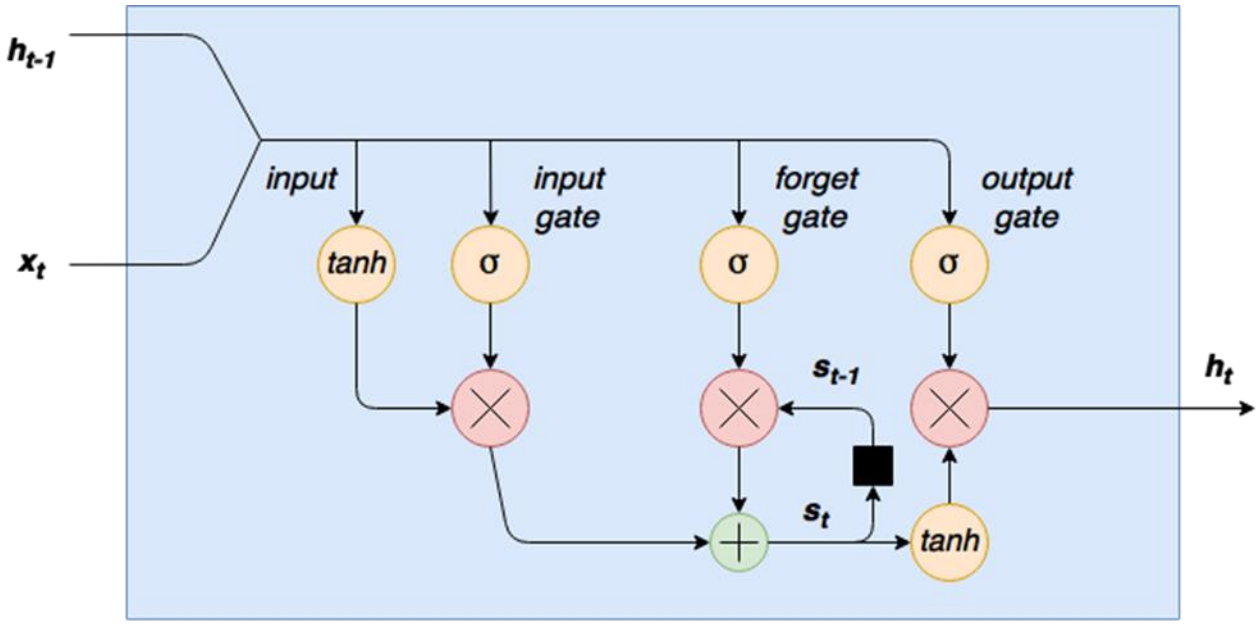


Recurrent neural networks are "unrolled" programmatically during training and prediction, so we get something like the following:



LSTM networks:

As mentioned previously, in this Keras LSTM tutorial we will be building an LSTM network for text prediction. An LSTM network is a recurrent neural network that has LSTM cell blocks in place of our standard neural network layers. These cells have various components called the input gate, the forget gate and the output gate – these will be explained more fully later. Here is a graphical representation of the LSTM cell:



Notice first, on the left hand side, we have our new word/sequence value txt being concatenated to the previous output from the cell ht−1ht−1. The first step for this combined input is for it to be squashed via a tan-h layer. The second step is that this input is passed through an input gate. An input gate is a layer of sigmoid activated nodes whose output is multiplied by the squashed input. These input gate sigmoid can act to "kill off" any elements of the input vector that aren't required. A sigmoid function outputs values between 0 and 1, so the weights connecting the input to these nodes can be trained to output values close to zero to "switch off" certain input values (or conversely, outputs close to 1 to "pass through" other values).

The next step in the flow of data through this cell is the internal state / forget gate loop. LSTM cells have an internal state variable st. This variable, lagged one time step i.e. st−1st−1 is added to the input data to create an effective layer of recurrence. This addition operation, instead of a multiplication operation, helps to reduce the risk of vanishing gradients. However, this recurrence loop is controlled by a forget gate – this works the same as the input gate, but instead helps the network learn which state variables should be "remembered" or "forgotten".

Input

First, the input is squashed between -1 and 1 using a *tanh* activation function. This can be expressed by:

$$g=tanh(b_g+x_tU_g+h_{t-1}V_g)\text{g=tanh(bg+xtUg+ht−1Vg)}$$

Where U_gUg and V_gVg are the weights for the input and previous cell output, respectively, and $b_g$bg is the input bias. Note that the exponents *g* are not a raised power, but rather signify that these are the input weights and bias values (as opposed to the input gate, forget gate, output gate etc.).

This squashed input is then multiplied element-wise by the output of the *input gate*, which, as discussed above, is a series of sigmoid activated nodes:

$$i=\sigma(b_i+x_tU_i+h_{t-1}V_i)\text{i=σ(bi+xtUi+ht−1Vi)}$$

The output of the input section of the LSTM cell is then given by:

$$g\circ i\text{g∘i}$$

Where the ∘ ∘ operator expresses element-wise multiplication.

Forget gate and state loop

The forget gate output is expressed as:

$$f=\sigma(b_f+x_tU_f+h_{t-1}V_f)\text{f=σ(bf+xtUf+ht−1Vf)}$$

The output of the element-wise product of the previous state and the forget gate is expressed as $s_{t-1}\circ f$st−1∘ f. The output from the forget gate / state loop stage is:

$$s_t=s_{t-1}\circ f+g\circ i\text{st=st−1∘f+g∘i}$$

Output gate

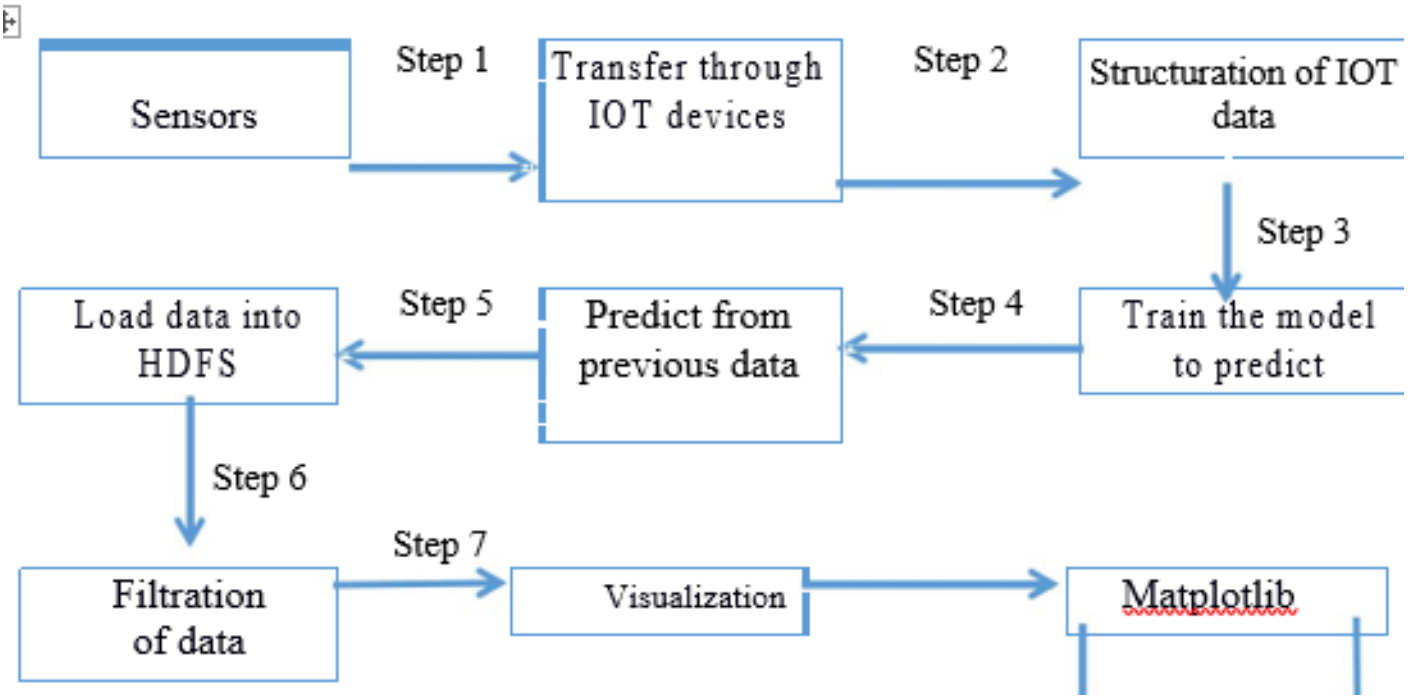
The output gate is expressed as:

$$o=\sigma(b_o+x_tU_o+h_{t-1}V_o)\text{o=σ(bo+xtUo+ht−1Vo)}$$

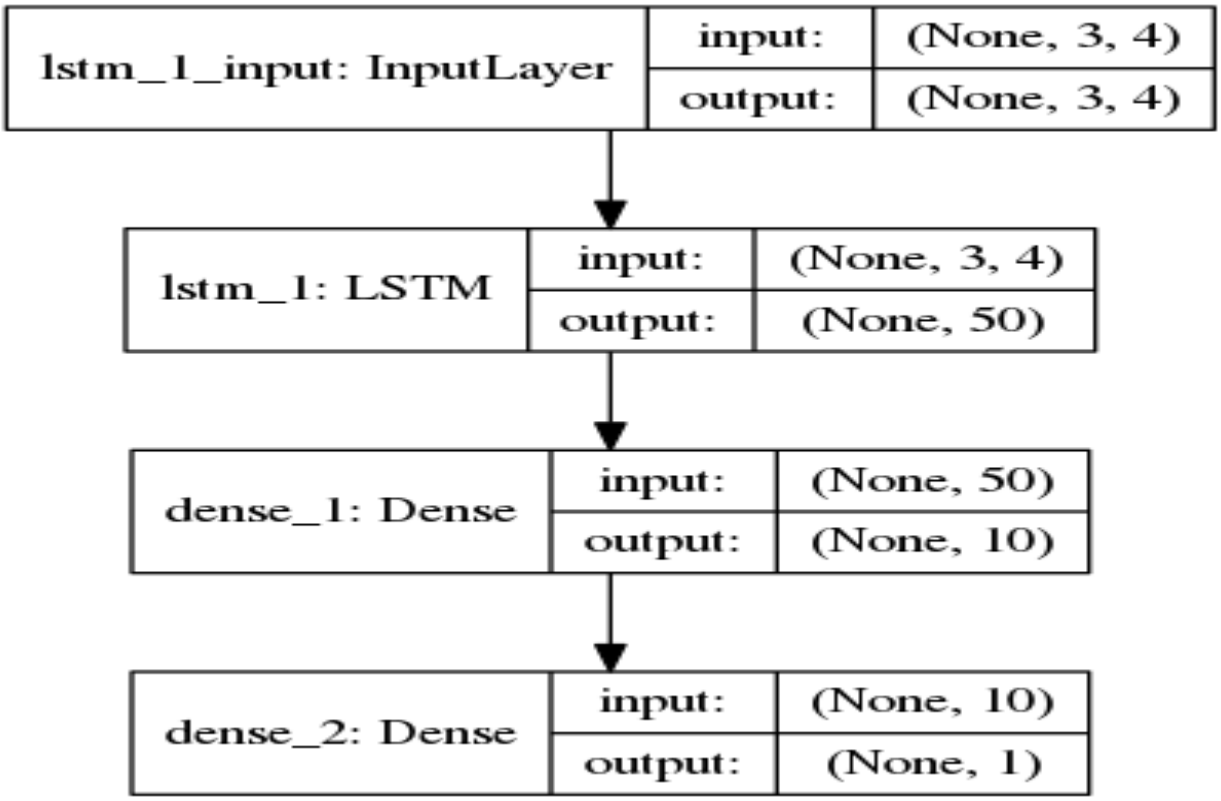
So the final output of the cell , with the *tanh* squashing, can be shown as:

$$h_t=tanh(s_t)\circ o$$

Implementation Details



Model layout



Workflow

Step Number	Step name	Description
1	Sensor Data	Sensors from 449 locations are sending data about ozone, SO_2, NO_2, Particulate Matter to 449 primary data hubs.
2	Transfer through IOT Device	The Sensor Data goes to the main station through the IOT device for Prediction.
3	Structuration of Data	Once the raw data have arrived to main station from those 449 flume agents the spark-streaming collects them make them structured and them those structured files Are saved into different spool directories.
4	Learn to Predict	The very next step involves machine learning through Neural Networks. These networks are running across distributed platform and giving predictions for Max AQI value, 8am-11am time range value, 5pm-8pm time range values of different locations.
5	Filtration	From these values we are discarding those values which are less than 150 which is a good AQI Zone. We are only concerned about hazardous values.
6	Visualization	Using Matplotlib to visualize the predicted pattern.

Results/Sample output

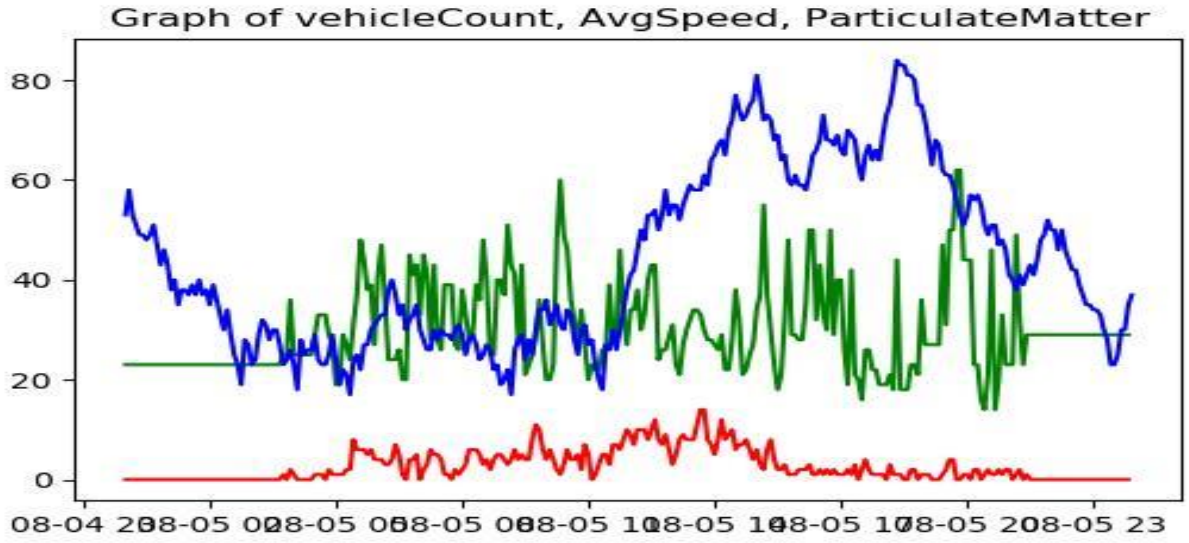
Dataset for AQI of ozone, particulate matter, CO, NO2, SO2

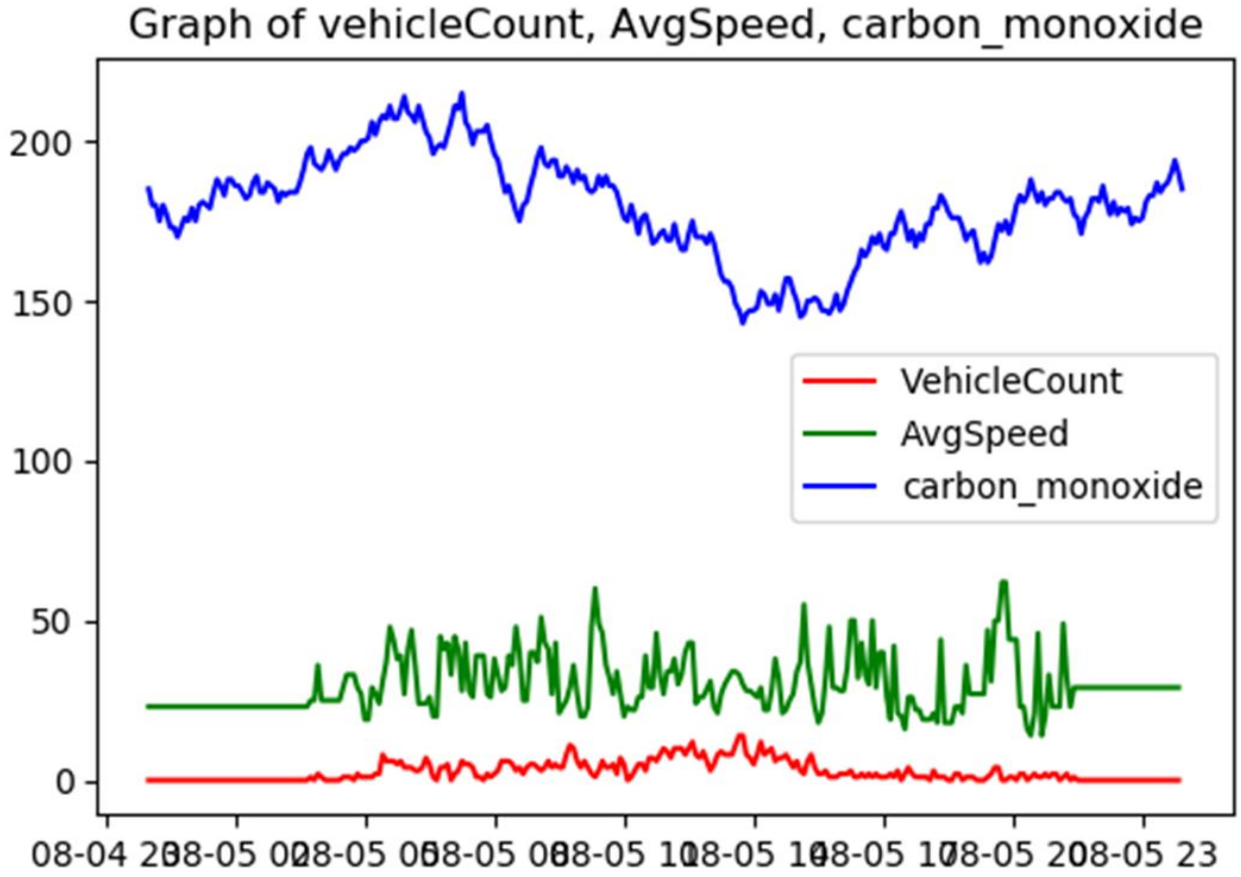
http://iot.ee.surrey.ac.uk:8080/datasets.html

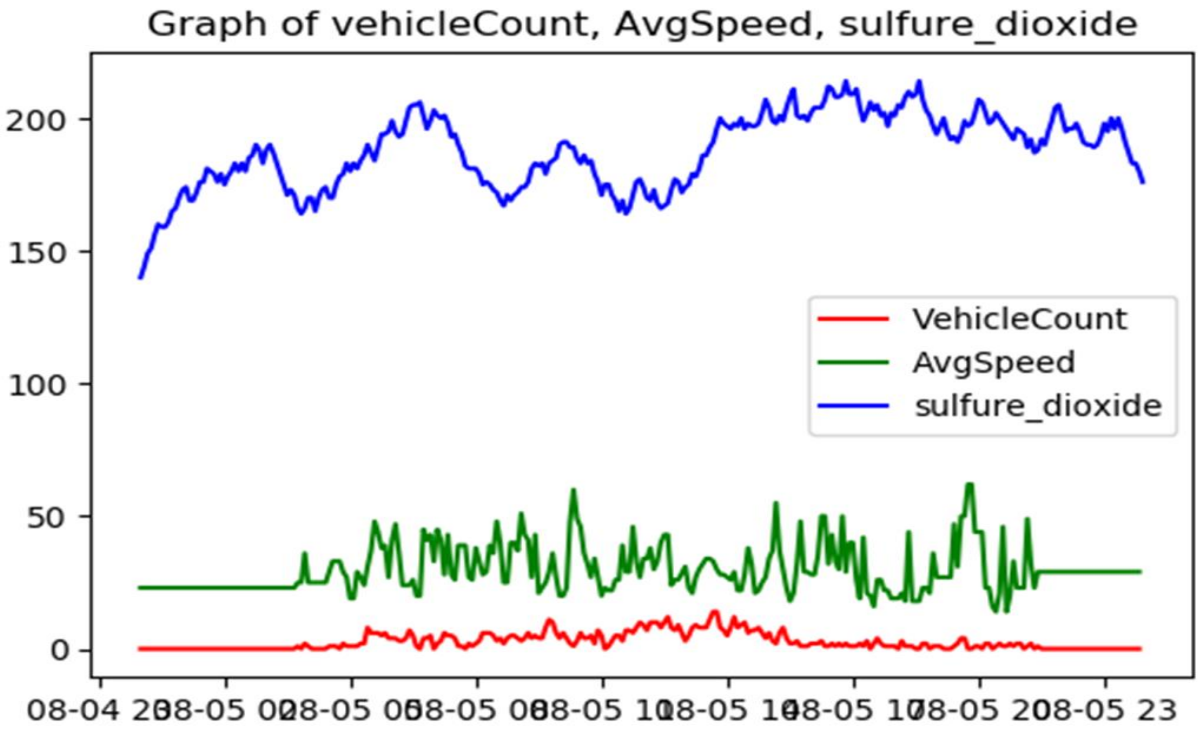
	ozone	particullate_matter	carbon_monoxide	sulfure_dioxide	nitrogen_dioxide	longitude	latitude	timestamp
2	29	37	27	60	87	10.18936	56.1821	8/1/2014 8:00
3	31	32	28	62	91	10.18936	56.1821	8/1/2014 8:05
4	29	37	33	62	93	10.18936	56.1821	8/1/2014 8:10
5	27	40	35	61	96	10.18936	56.1821	8/1/2014 8:15
6	22	35	39	65	92	10.18936	56.1821	8/1/2014 8:20
7	24	33	36	68	97	10.18936	56.1821	8/1/2014 8:25
8	29	35	34	71	95	10.18936	56.1821	8/1/2014 8:30
9	33	35	37	70	90	10.18936	56.1821	8/1/2014 8:35
10	32	32	35	74	94	10.18936	56.1821	8/1/2014 8:40
11	35	36	35	77	94	10.18936	56.1821	8/1/2014 8:45
12	39	40	30	74	96	10.18936	56.1821	8/1/2014 8:50
13	38	42	30	70	95	10.18936	56.1821	8/1/2014 8:55
14	41	45	33	68	90	10.18936	56.1821	8/1/2014 9:00
15	37	43	36	64	95	10.18936	56.1821	8/1/2014 9:05

Dataset for Average Speed, Average measured time, Vehicle count on different timestamp

	A	B	C	D	E	F	G	H	I
1	status	avgMeasuredTime	avgSpeed	extID	medianMeasuredTime	TIMESTAMP	vehicleCount	_id	REPORT_ID
2	OK	51	41	727	51	2014-08-01T08:00:00	1	20746782	204273
3	OK	90	23	727	90	2014-08-01T08:05:00	3	20747231	204273
4	OK	71	29	727	71	2014-08-01T08:10:00	3	20747604	204273
5	OK	69	30	727	69	2014-08-01T08:15:00	6	20748053	204273
6	OK	49	43	727	49	2014-08-01T08:20:00	4	20748502	204273
7	OK	43	49	727	43	2014-08-01T08:25:00	7	20748951	204273
8	OK	40	53	727	40	2014-08-01T08:30:00	7	20749400	204273
9	OK	101	21	727	101	2014-08-01T08:35:00	6	20749849	204273
10	OK	94	22	727	94	2014-08-01T08:40:00	3	20750298	204273
11	OK	42	50	727	42	2014-08-01T08:45:00	5	20750747	204273
12	OK	42	50	727	42	2014-08-01T08:50:00	5	20751196	204273
13	OK	71	29	727	71	2014-08-01T08:55:00	4	20751645	204273
14	OK	68	31	727	68	2014-08-01T09:00:00	7	20752094	204273


Graph of vehicleCount, AvgSpeed, ParticulateMatter
80
60
40
20
0
08-04 2
08-05 0
08-05 0
08-05 0
08-05 1
08-05 1
08-05 1
08-05 2
08-05 23


Graph of vehicleCount, AvgSpeed, carbon_monoxide
200
150
100
50
0
VehicleCount
AvgSpeed
carbon_monoxide
08-04 2
08-05 0
08-05 0
08-05 0
08-05 1
08-05 1
08-05 1
08-05 2
08-05 23



Result

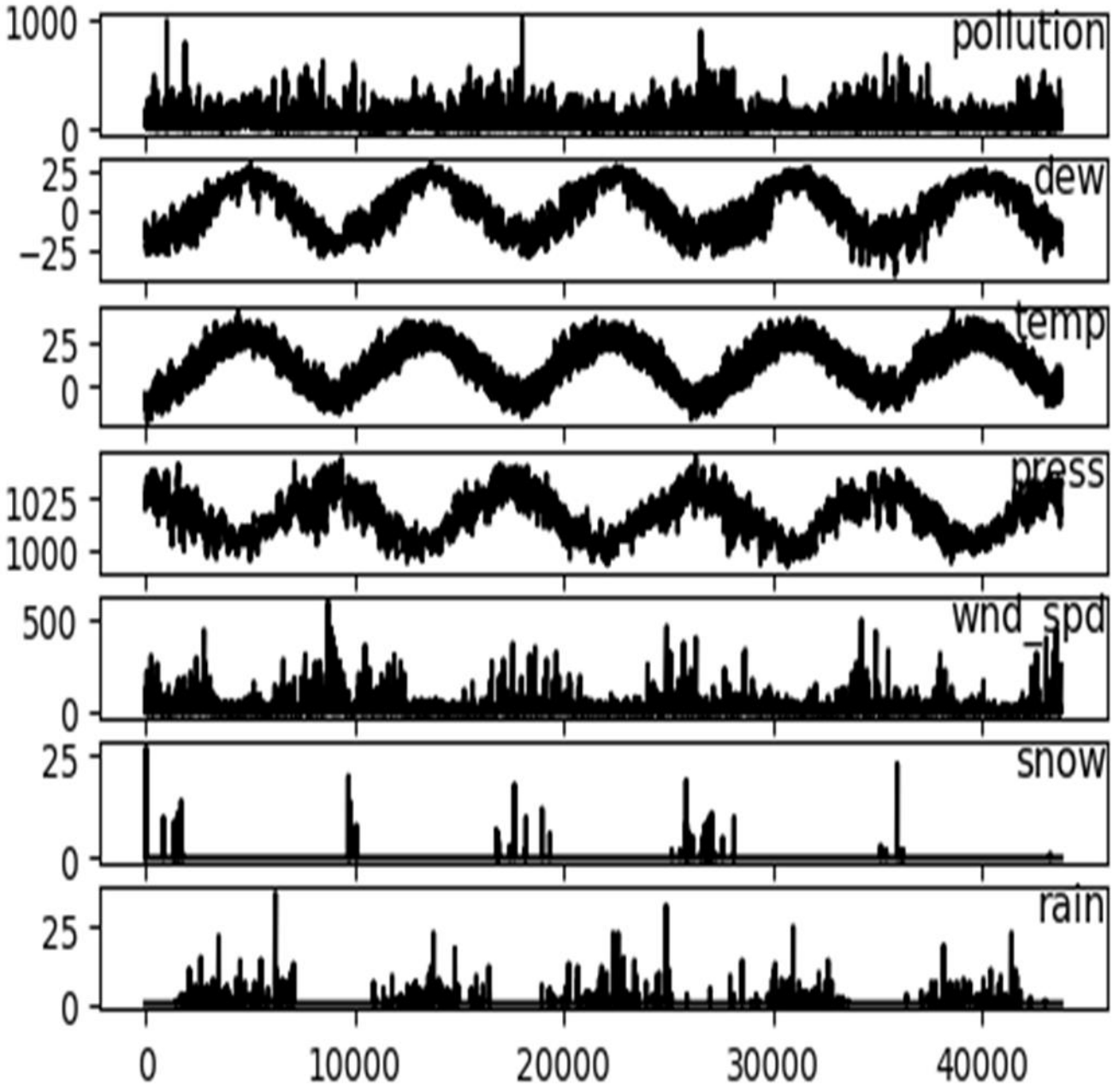
For this dataset of Denmark form we tried to predict the value of particulate matter, SO2, NO2, Ozone based on vehicle count, vehicle speed and get about 26% accuracy as the correlation among those variable was very less(not more than 10%)

Dataset 2 Pollution data(date,pollution,dew,temp,press,wnd_dir,wnd_spd,snow,rain)

https://github.com/sagarmk/Forecasting-on-Air-pollution-with-RNN-LSTM/blob/master/pollution.csv

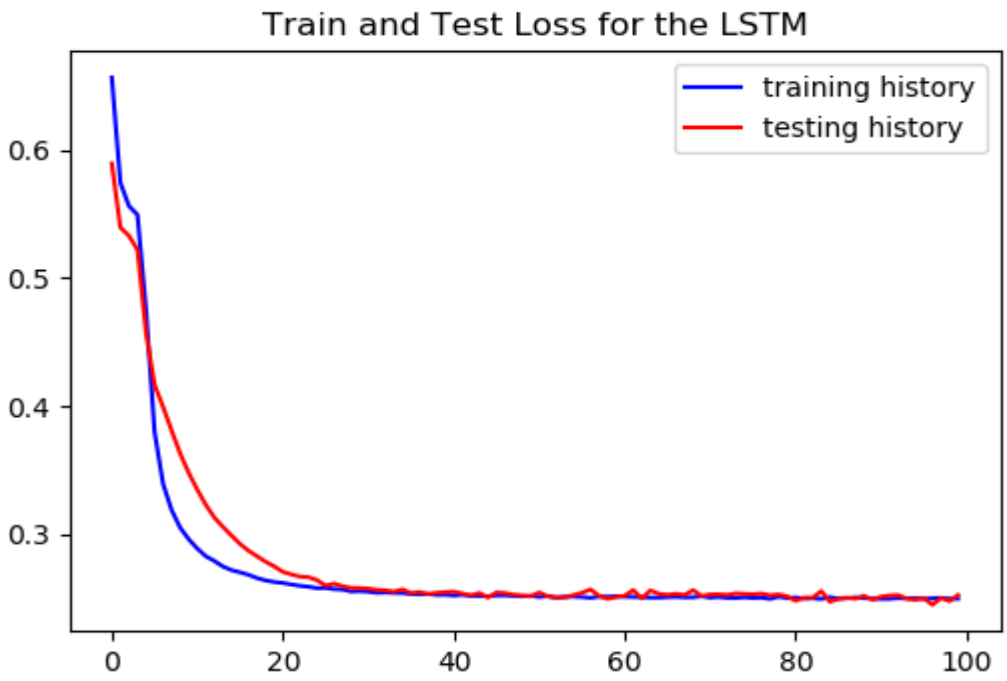
	A	B	C	D	E	F	G	H	I
1	date	pollution	dew	temp	press	wnd_dir	wnd_spd	snow	rain
2	2010-01-02 00:00:00	129	-16	-4	1020	SE	1.79	0	0
3	2010-01-02 01:00:00	148	-15	-4	1020	SE	2.68	0	0
4	2010-01-02 02:00:00	159	-11	-5	1021	SE	3.57	0	0
5	2010-01-02 03:00:00	181	-7	-5	1022	SE	5.36	1	0
6	2010-01-02 04:00:00	138	-7	-5	1022	SE	6.25	2	0
7	2010-01-02 05:00:00	109	-7	-6	1022	SE	7.14	3	0
8	2010-01-02 06:00:00	105	-7	-6	1023	SE	8.93	4	0
9	2010-01-02 07:00:00	124	-7	-5	1024	SE	10.72	0	0
10	2010-01-02 08:00:00	120	-8	-6	1024	SE	12.51	0	0
11	2010-01-02 09:00:00	132	-7	-5	1025	SE	14.3	0	0
12	2010-01-02 10:00:00	140	-7	-5	1026	SE	17.43	1	0
13	2010-01-02 11:00:00	152	-8	-5	1026	SE	20.56	0	0
14	2010-01-02 12:00:00	148	-8	-5	1026	SE	23.69	0	0
15	2010-01-02 13:00:00	164	-8	-5	1025	SE	27.71	0	0
16	2010-01-02 14:00:00	158	-9	-5	1025	SE	31.73	0	0
17	2010-01-02 15:00:00	154	-9	-5	1025	SE	35.75	0	0
18	2010-01-02 16:00:00	159	-9	-5	1026	SE	37.54	0	0
19	2010-01-02 17:00:00	164	-8	-5	1027	SE	39.33	0	0
20	2010-01-02 18:00:00	170	-8	-5	1027	SE	42.46	0	0
21	2010-01-02 19:00:00	149	-8	-5	1028	SE	44.25	0	0
22	2010-01-02 20:00:00	154	-7	-5	1028	SE	46.04	0	0
23	2010-01-02 21:00:00	164	-7	-5	1027	SE	49.17	1	0
24	2010-01-02 22:00:00	156	-8	-6	1028	SE	52.3	2	0
25	2010-01-02 23:00:00	126	-8	-6	1027	SE	55.43	3	0
26	2010-01-03 00:00:00	90	-7	-6	1027	SE	58.56	4	0
27	2010-01-03 01:00:00	63	-8	-6	1026	SE	61.69	5	0
28	2010-01-03 02:00:00	65	-8	-7	1026	SE	65.71	6	0
29	2010-01-03 03:00:00	55	-8	-7	1025	SE	68.84	7	0
30	2010-01-03 04:00:00	65	-8	-7	1024	SE	72.86	8	0

pollution



Graph of pollution data (Dataset 2)

When a dataset is trained and then tested with LSTM, a loss or error occurs because of the huge data.



Result 2:

In this dataset we get about 82% accuracy in timestep2 and epoch 50. We have tried with more epoch and more timestep and also added more layer but the result was the best for the above implementation.

Conclusion

Air pollution is a serious environmental concern all around the globe. Over the last few decades, the intensified process of industrialization and urbanization, coupled with rapid population growth has resulted in severe environmental degradation. In particular, harmful pollutants such as Sulphur Dioxide (SO2), Nitrogen Dioxide (NO2), Ozone (O3), Total Suspended Particles Matter (TSPM) etc, are emitted and these pollutants even exceed air quality guidelines recommended by the World Health Organization (WHO, 2005).Particulate and gaseous emissions of pollutants from industries and auto exhaust are responsible for rising discomfort, increasing airborne diseases, decreasing productivity and deterioration of artistic and cultural patrimony urban center. We have analyzed the air quality index and prepared for pre-diction. Our Future goal is to regulate traffic by making adjustments in the signal durations at various intersections and improve the predictive model for better performance.

References

Keras documentation: www.keras.io

Time Series Prediction: http://machinelearningmastery.com

Neural Network Models for Air Quality Prediction: A Comparative Study By S V Barai, A K Dikshit , Sameer Sharma

[1999]Applying machine learning techniques in air quality prediction by Elias Kalapanidas and Nikolaos Avouris

[Abdel 1996] Abdel-Aal R. E.; Elhadidy M. A. (1996). Modelling and forecasting the daily maximum temperature using abductive machine learning. Oceanographic Literature Review, 43 (1).

Probability and Statistics for Programmers Version 1.6.0 By Allen B. Downey

[sklearn http://scikit-learn.org/stable/documentation.html]

[2009] Recurrent neural network for air pollution peaks prediction for the region of Annaba – Algeria by Ghazi sabri1 Khadir Mohamed Tarek

[2002]INTEGRATION OF TRAFFIC MANAGEMENT AND AIR QUALITY CONTROL (iTRAQ) By Stefan Gustafsson, Norbert Hübner, Benjamin N. Passow, David Elizondo, Eric Goodyer, Yingjie Yang, Roland J. Leigh, James P. Lawrence, Satish Shah, Jolanta Obszynska, Sarah C.M. Brown, Andrew Groom

Fundamentals of Neural Networks: Architectures, Algorithms and Applications, 1e By Fosset
[2015]Air Quality Data Analytics using Spark and ESRI's GIS Tools by Brett Gaines and Qi Dai