Air quality particulate-pollution prediction applying GAN network and the neural Turing machine

[1, 2] Zahra-Sadat Asaei-Moamam, [1,2]Faramraz Safi-Esfahani (corresponding author), [3]Seyedali Mirjalili,
[4]Reza Mohammadpour, [1,2]Mohamad-Hosein Nadimi-Shahraki
z.asaei61@gmail.com, fsafi@iaun.ac.ir, reza564@gmail.com, nadimi@iaun.ac.ir
[1]Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran.
[2]Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran.
[3]Centre for Artificial Intelligence Research and Optimisation, Torrens University, Australia.
[4]Department of civil engineering, Estahban Branch, Islamic Azad University, Estahban, Iran.

Abstract

Urban areas in many countries face significant concerns over the presence of aerosol particles and their effects on human health. These particles, which range in size from 1 nanometer to 100 micrometers, can easily penetrate organic matter and transport toxic gas compounds and mineral substances such as carbon monoxide, ozone, nitrogen dioxide, and sulfur dioxide. High concentrations of airborne particles pose serious challenges to health, the economy, the environment, and society, and it is crucial to investigate ways to improve the quality of life for people. The main objective of this study is to develop a framework that accurately predicts aerosol and air quality index (AQI) values in advance by estimating missing data and predicting future data. To achieve this, we propose the DAerosol.GAN.NTM framework, which combines a neural Turing machine with a generative adversarial network (GAN) to address the limitations of previous studies. Our framework outperforms previous methods, including DAerosol.NTM (without GAN) and four baseline studies using Multilayer Perceptron (MLP), Deep Neural Networks (DNN), Long-short Term Memory (LSTM), and Deep LSTM (DLSTM) models, in terms of accuracy, precision, root mean square error (RMSE), mean absolute percentage error (MAPE), and the prediction of aerosol pollution surge hours in advance according to the Time Interval Before and After the Aerosol Event (TIBAAE) criterion.

Keywords: Neural Turing Machines (NTM); Generative Adversarial Networks (GAN); Deep Learning (DL); *Air Quality Index (AQI);* Particulate Matter 2.5 ($PM_{2.5}$); Particulate Matter 10 (PM_{10}); Aerosol;

1 Introduction

Particulate matter, also known as aerosol, consists of fine airborne particles ranging from approximately 1 nanometer to 100 micrometers(Qin et al., 2010)(Nakata et al., 2015). These particles are large enough to carry toxic gaseous compounds, minerals, and infectious diseases, but small enough to penetrate lung tissue and potentially carry harmful elements into the body, eliciting harmful reactions. The Air Quality Index (AQI) is a widely used indicator that considers the principal parameters that shape air quality and safety. This index relies daily on six air pollutants: carbon monoxide (CO), ozone (O_3), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), particulate matter 2.5 microns ($PM_{2.5}$), and particulate matter 10 microns (PM_{10})(Zhu et al., 2017) (Diro & Chilamkurti, 2018). The pollutants with the highest concentration ultimately determine the AQI value, among which $PM_{2.5}$ and PM_{10} are usually leading and evaluated by twenty-two meteorological parameters plus CO, O3, NO2, and SO2. The AQI index is reported on a continuous scale of 0 to 500 and divided into six pollution categories for the convenience of public announcements so that they show the impact of air quality on human health and provide a straightforward numerical reference for risk management of outdoor activities. It also provides better design ideas for the expansion of cities. Previous attempts have been made to estimate and forecast the principal parameters of the AQI or the AQI value itself using machine learning and neural network methods and compare the performance of different approaches. Firstly (S. Kim et al., 2019) and (Xayasouk et al., 2020) implemented and compared frameworks using Deep Autoencoder (DAE), Deep Neural Networks (DNN), and Deep Long Short-Term Memory (DLSTM) in forecasting AQI parameters in 24-hour (daily) time intervals, namely $PM_{2.5}$ and PM_{10}, in the Seoul Metropolitan Area. Data were collected at 25 pollution and weather stations. The DLSTM method reduced the error rate of the DAE and DNN models due to a lack of normalization and transparent data processing. Both studies applied to meteorological data from a single meteorological station, which was considered insufficient to predict aerosol parameters in a widely used district of Seoul. In addition, evaluating only the $PM_{2.5}$ and PM_{10} from among six air pollutants is insufficient in acquiring a holistic view of the AQI index either.

The article (Sharma et al., 2018) differs from previous seasonal trends in AQI parameters. It implements many more time series data from India's Pollution Control Centre. The data estimate the value of the entire AQI (rather than its components). The article uses a deep neural network architecture in which Recurrent Neural Networks (RNN) and DLSTM are compared with the Autoregressive Integrated Moving Average (ARIMA)(Ma et al., 2019). The DLSTM has once again surpassed the other models, this time with the advantage of being insensitive to the length of the interval. Similar to the above baseline methods, this framework did not use meteorological data from numerous districts without clear preprocessing and data normalization.

The approach(Pengfei et al., 2018a) applies a DNN model after precise data processing and normalization to forecast $PM_{2.5}$ in the Wuhan district. Super Vector Machine (SVM) models and a traditional backpropagation network with three hidden layers were applied to analyze data. The results supported that the DNN model was superior to shallow learning methods in prediction, error consideration, and regression rate calculation. This study does not apply meteorological data and uses only pollution data from Wuhan. The selected model was not suitable for time series data because of a lack of memory that resulted in weight loss.

Ultimately all baseline studies (S. Kim et al., 2019)(Xayasouk et al., 2020)(Sharma et al., 2018)(Pengfei et al., 2018b) suffer from several shortcomings are detailed as follows: 1) Not using both meteorological and local pollution parameters as effective indicators in estimating air quality, 2) Using a single meteorological station for a big city and not to have local meteorological stations in different areas, 3) Low model accuracy due to not to keep data history and not to have long-term memory in the models, and 4) The failure to develop a suitable model for estimating lost data and producing data for the coming days.

To overcome the deficits above and to accurately predict aerosol and AQI indices ahead of time accordingly, the main task of this study is to provide a framework that can estimate the lost data and predict future data on time. Twenty-six independent variables exist, of which twenty-two are meteorological data, and four describe air quality. A high-level view of the independent variables includes the amount of carbon monoxide, CO, ozone O3, nitrogen dioxide NO2, sulfur dioxide SO2, sky condition, horizontal visibility, pressure, wind, relative humidity, and temperature. The dependent variables in this study are accuracy, precision, root mean square error (RMSE), and mean absolute percentage error (MAPE)(Li et al., 2016)(Kök et al., 2017) in terms of suspended particle matter size $PM_{2.5}$, PM_{10}, and AQI quality control index. The TIBAAE parameter which is the Time Interval Before and After Aerosol Event has also been introduced.

The first research hypothesis states that using an external memory in the deep learning neural network improves the accuracy and precision of $PM_{2.5}$, PM_{10}, and the AQI quality control index due to the long history of meteorological information. The second hypothesis of the research states that if the GAN-based layer is used to estimate the lost data and the data of the coming days together with the NTM network, it will improve results to predict the particle size $PM_{2.5}$, PM_{10}, and AQI. This study aims to enhance people's quality of life by focusing on aerosol forecasting. In this study, a framework called DAerosol.GAN.NTM (Deep Aerosol – Generative Adversarial Networks - Neural Turing Machines) predicts air quality parameters using deep learning. Ultimately, DAerosol.GAN.NTM hopes to improve our understanding of aerosol pollution in terms of the AQI by maximizing our usage of available meteorological data and modern framework approach such as NTM and GAN to outperform previous baseline models and aid in the mitigation of otherwise harmful weather events that could have considerable socio-economic and public health state of any urban environment.

The rest of the document is organized as follows: Section 2 looks at related work. The framework elaborated in this study is presented in Section 3. Section 4 describes the experimental design. Section 5 suggests future research directions and Section 6 concludes by addressing the theoretical concepts.

2 Literature Review

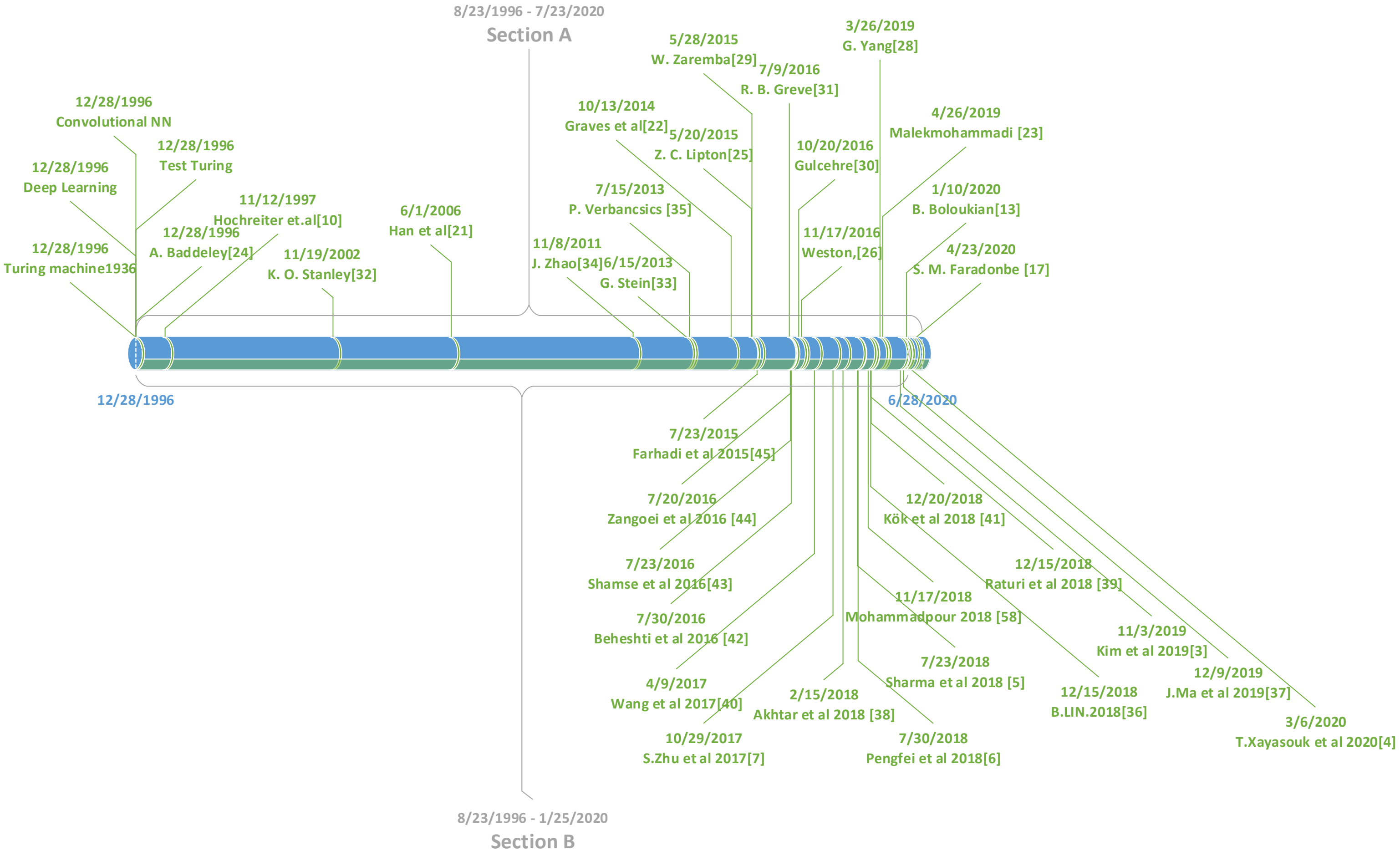


Figure 1: Research timeline for predicting air quality parameters.

Figure 1 presents the chronological literature in sections A and B. The former section includes the literature on the neural Turing machine (NTM), and the latter consists of research on predicting air quality parameters.

2.1 Section A - Neural Turing Machine (NTM)

Research on intelligent machines dates back more than 70 years. Simultaneously with the creation of the first electronic computers(Gulcehre et al., 2016), the design of algorithms has always been one of the artificial intelligence's goals (Turing, 1950). One of the first attempts to create an intelligent machine was when Alan Turing invented the Turing machine in 1936. Initially, the Turing program significantly impacted psychology and cognitive philosophy. These contexts saw the computer as a metaphor for the brain's functional model. This metaphor was quickly dismissed by neuroscience due to the dissimilarity of the Turing machine architecture compared to humans (Siegelmann & Sontag, 1991).

AI means machine intelligence or the ability to perceive or learn a machine. The machine here means any smart device that has a processor. It can take input data from the environment, process it, and possibly make decisions. Artificial intelligence's primary and ultimate goal is to simulate and understand human behavior. Machine learning, the core of artificial intelligence, involves various methods and algorithms that work differently. The first examples of convolutional neural networks were presented in the 1990s. In 1993, the first successful example of a convolutional neural network was designed. A few years later, recurrent neural networks, or RNNs, were developed called LSTM. They suffered from limited input data, required powerful hardware, and had training problems.

The year 2006 was called the year of overcoming this crisis. Because this year, a new method of teaching deep networks was presented (Han et al., 2006). By 2014, many scientists worked in this field, and the number of articles and books was growing. From 2012 to 2020, the number of projects proposed in deep learning increased rapidly. NTM was presented for the first time by Grave et al. in 2014(Graves et al., 2014). A neural network can read and write on an external memory matrix, similar to random memory on a regular computer. The NTM neural network can use its memory to display and manipulate complex data structures. At the same time, the same neural network can learn from the data. The following section reviews the related works on neural Turing machines.

Many studies have been conducted using machine learning and deep learning methods in the last decade (Graves et al., 2014)(Malekmohammadi Faradonbeh et al., 2019). The research (Baddeley, 1996) employed a Turing machine and a neural network analogous to combining RNN recursive neural networks with external memory to perform algorithmic tasks. Compared to the Turing machine, the NTM method of a neural controller directs the read and write heads to the external memory as a tape(Lipton et al., 2015). In the paper (Weston et al., 2016), a machine-learning model derived from NTM alters the memory architecture and presents a DNC[1] separable neural computer model. This model uses the same architecture as the neural network controller with read-write head access to the memory matrix. The access mechanism and its relationship to memory are different. Information is retained if it is repeated a certain number of times.

Supervised learning has been shown to answer compound questions designed to mimic problems and inference in natural language (using the bAbI database). In the article (Graves et al., 2016), a detachable neural computer can find the shortest path between specified points (such as the transport network). It can also guess missing links (such as family genealogies) in randomly generated graphs. Other methods that have been proposed so far have made changes to the NTM method controller. These include Lie Access NTM(Yang & Rush, 2019), RL NTM(Zaremba & Sutskever, 2015), and Dynamic NTM(Gulcehre et al., 2018), which are used for copying and pasting tasks.

Nevertheless, they have not been improved to solve more complex tasks in Neuro-evolution and do not have enough accuracy and speed. The paper(Greve et al., 2016) has shown that the neural Turing machine reinforcing the ENTM[2] topology is an evolutionary version of NTM. The NTM method was developed with the neural evolution algorithm to strengthen the NEAT[3] topology (Stanley & Miikkulainen, 2002). NEAT is a striking example of a genetic algorithm(Stein et al., 2013). Unlike traditional methods, ANN topology does not need to be designed. Instead, it can create and transform the ANN topology and weight (Zhao & Peng, 2011). The NEAT method starts with a small population of simple neural networks. Adding new nodes and connections through the mutation increases its complexity. In general, NEAT is an evolutionary method that finds the correct topology and weight of a network to maximize the performance of a task (Verbancsics & Harguess, 2013). The ENTM method(Greve et al., 2016) also can solve a simple task such as copying. For the first time, the trial version of double (T-Maze) raises an issue of reinforcement learning. In the learning task (T-Maze), the agent uses memory to adapt to behavior (usually requires a neural network). The common problems of this method are high computation and memory limitation.

Therefore, the practical choice of a model for predicting air quality parameters can be the strength of this study. It is used to prevent deep network problems that lead to weight retention and disappearance during the upgrade.

2.2 Section B - Forecast of air quality parameters

Over recent years, many research works have focused on the impact of air pollution on human health and the environment(Liang et al., 2019). The main goal of the present study is to predict air quality parameters from 2015 to 2020. Air quality information is vital for protecting human health and controlling air pollution (Sharma et al., 2018). In recent years, there has been a significant increase in the number of vehicles and factories. This trend is expected to continue soon. Air pollution also has a significant impact on architecture and the environment.

Moreover, air pollution causes water pollution in rivers and lakes, which hurts the health of animals and plants(Ajibade et al., 2020). Environmental problems are becoming more severe due to overpopulation in cities (Lin & Zhu, 2018). Researchers have different views on the relationship between urbanization and the urban environment(Zhu et al., 2017)(Appannagari, 2006)(Lovett et al., 2009). Some believe that increasing urbanization exacerbates urban environmental pollution(Lu et al., 2021). Urbanization has made fundamental changes in human life. First, people use natural resources to produce the

[1] Differentiable Neural Computer (DNC)

[2] Evolving Neural Turing Machine (ENTM)

[3] Neuro Evolution of Augmenting Topologies (NEAT)

necessary products and make many pollutants. Second, urbanization has created a demand for transportation and infrastructure systems, increasing household energy consumption. Rising energy consumption causes the production of many pollutants and ultimately worsens the urban environment (Lin & Zhu, 2018)(Liang et al., 2019). Figure 2 displays the mind map of this study and presents a brief review of the methods that have been used so far to predict the air quality parameters.

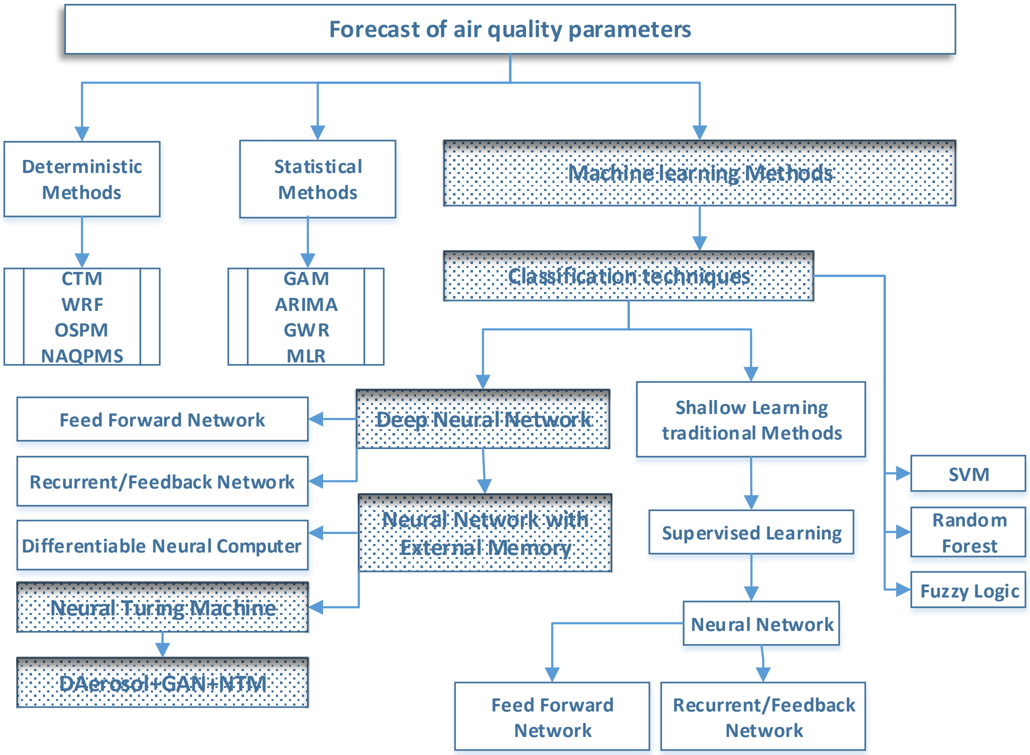


Figure 2: Mind-Map diagram and the mainstream of this research among the baseline methods

Figure 2 illustrates that research on the prediction of air quality parameters can be divided into three sections deterministic, statistical, and machine learning (Rybarczyk & Zalakeviciute, 2018).

The deterministic branch includes CTMs[4], WRF-Chem[5], OSPM[6], and NAQPMS[7] methods that construct models to simulate atmospheric chemistry dispersion and transfer. In statistical methods (e.g., GAM[8], ARIMA[9], GWR[10], and MLR[11]), the concentration of pollutants is observed using a large amount of data. These methods overcome the limitations of deterministic methods but suffer from statistical limitations. Machine learning has applied SVM[12], ANNs[13], FL[14], and RF[15] in predicting air quality factors, among which are the most famous neural networks (Ma et al., 2019).

[4] Chemical Transport Models (CTMs)
[5] Weather Research and Forecasting (WRF)model coupled with Chemistry (Chem)
[6] Operational Street Pollution Models (OSPM)
[7] Nested Air Quality Prediction Modeling System (NAQPMS)
[8] Generalized Additive Models (GAMs),
[9] Autoregressive Integrated Moving Average (ARIMA)
[10] Geographically Weighted Regression (GWR)
[11] Multi-layer Regression (MLR)
[12] Support Vector Machine (SVM)
[13] Artificial Neural Networks (ANNs)
[14] Fuzzy Logic (FL)
[15] Random Forest (RF)

The highlighted area shows the novelty of the research by applying external memory, such as NTM, and its combination with the GAN network. The proposed method uses a deep learning method, combining the Turing machine and LSTM recursive neural networks with an addressable external memory and a GAN network to predict the air quality parameters before the occurrence of aerosol pollution. Table 1 compares the methods used in the related works and the present study.

Table 1: Comparison of the most related research works on aerosol pollution prediction.

Research work	Data	Size of dataset			No. of independent variables	Method	Evaluated parameters	Accuracy	Precision	RMSE	MAPE	Removing missing data	The time interval before and after the aerosol event (TIBAAE)
		Period	Station number	Total sample size									
(Xayasouk et al., 2020)	Quality and Meteorological Data (Seoul)	January 1, 2015, to December 31, 2018	25 Quality Control and One Meteorological station	-	12	DLSTM, DAE	$PM_{2.5}$, PM_{10}	For DLSTM: $PM_{2.5}$=88%, PM_{10}=89% For DAE: $PM_{2.5}$=84%, PM_{10}=85%		For DLSTM: $PM_{2.5}$=11% PM_{10}=11% For DAE: $PM_{2.5}$=15%, PM_{10}=15%	-	-	-
International Conference **(S. Kim et al., 2019)**	Quality and Meteorological Data (Seoul)	One year	39 Quality Control and one Meteorological station	-	17	DNN, DLSTM	$PM_{2.5}$, PM_{10}	-	-	For DNN: $PM_{2.5}$=6%, PM_{10}=6% For DLSTM: $PM_{2.5}$=5%, PM_{10}=5%	-	-	-
(Pengfei et al., 2018b)	Quality Data (Wuhan)	-	One Quality Control Station		8	DNN, SVM, BP	$PM_{2.5}$	For DNN: $PM_{2.5}$= 90% For SVM: $PM_{2.5}$ = 66% For BP: $PM_{2.5}$= 67%	-	For DNN: $PM_{2.5}$= 9% For SVM = - For BP = -	-	-	-
(Sharma et al., 2018)	Quality Data (RK Puram New Delhi)	Two years	Quality Control Station	-	7	DLSTM	AQI	For DLSTM: AQI= 63%	-	For DLSTM: AQI= 37%	-	-	-
(Akhtar et al., 2018)	Quality Data Center (CPCB))	Two years	Quality Control Station		8	MLP	PM_{10}	For MLP: PM_{10}= 95%	-	-	-	-	-
(Wang et al., 2017)	Quality Data (China) http://113.108.142.147:20035/emcpublish/	Two years	-	-	6	ANFIS, AHP	AQI	For ANFIS: AQI=76%, For AHP: AQI=68%	-	-	-	-	-
International Conference **(Shams et al., 2017)**	Quality and Meteorological Data (Tehran)	-	Ten stations	-	6	Multivariate regression	AQI	For Multivariate regression: AQI =76%	-	-	-	-	-
(Zangouei et al., 2017)	Quality Data (Mashhad)	Two months	-	-	4	MLP	PM_{10}	For MLP: PM_{10}= 91%	-	-	-	-	-
<u>This study 2022</u>	**Quality and Meteorological Data (Tehran)**	**Twenty years**	**Four meteorological stations and fifteen air pollution sensors**	**58,922**	**29**	**DAerosol.GAN.NTM**	**$PM_{2.5}$ PM_{10} AQI TIBAAE**	$PM_{2.5}$**=96%,** PM_{10}**= 97%'** AQI = **82%**	$PM_{2.5}$**=87%,** PM_{10}**=88%,** AQI**=75%**	$PM_{2.5}$**=4%,** PM_{10}**=3%,** AQI **=18%**	$PM_{2.5}$**=13%** PM_{10}**=12%** AQI **=25%**	**Using GAN networks**	**$TIBAAE_{befor}$=96(H) $TIBAAE_{after}$=24(H)**

3 Presented approach: applying deep learning and neural Turing machine along with GAN network in aerosol prediction (DAerosol.GAN.NTM)

Figure 3 shows the framework proposed in this study: Deep Aerosol – Generative Adversarial Networks - Neural Turing Machines (DAerosol.GAN.NTM), which contains training, testing, and validation sections. This framework is used in GAN and NTM networks. Since the GAN network is applied to produce new data, delivering new intelligence in the "real data" learning section shows the labeled data. Training data includes both actual and unsupervised data generated by the GAN network. Applying a little noise makes the model more resistant to extracting better features. The data generated by the GAN network replace the corrupted data to train the main network better than when the network applies the original data. The results presented in the experimental section indicate an improvement compared to the findings reported in the proposed approaches in the baseline articles.

The primary purpose of using the LSTM recursive neural network in the NTM network controller is to analyze time-series data correctly and accurately. Each training step uses external memory to store and retrieve network weights to/from the NTM external memory. In other words, these weights are stored in the NTM memory with a specific coefficient in each training step. This memory makes a semantic correlation between different training iterations. During each update cycle, new weights are calculated from the weights stored in memory.

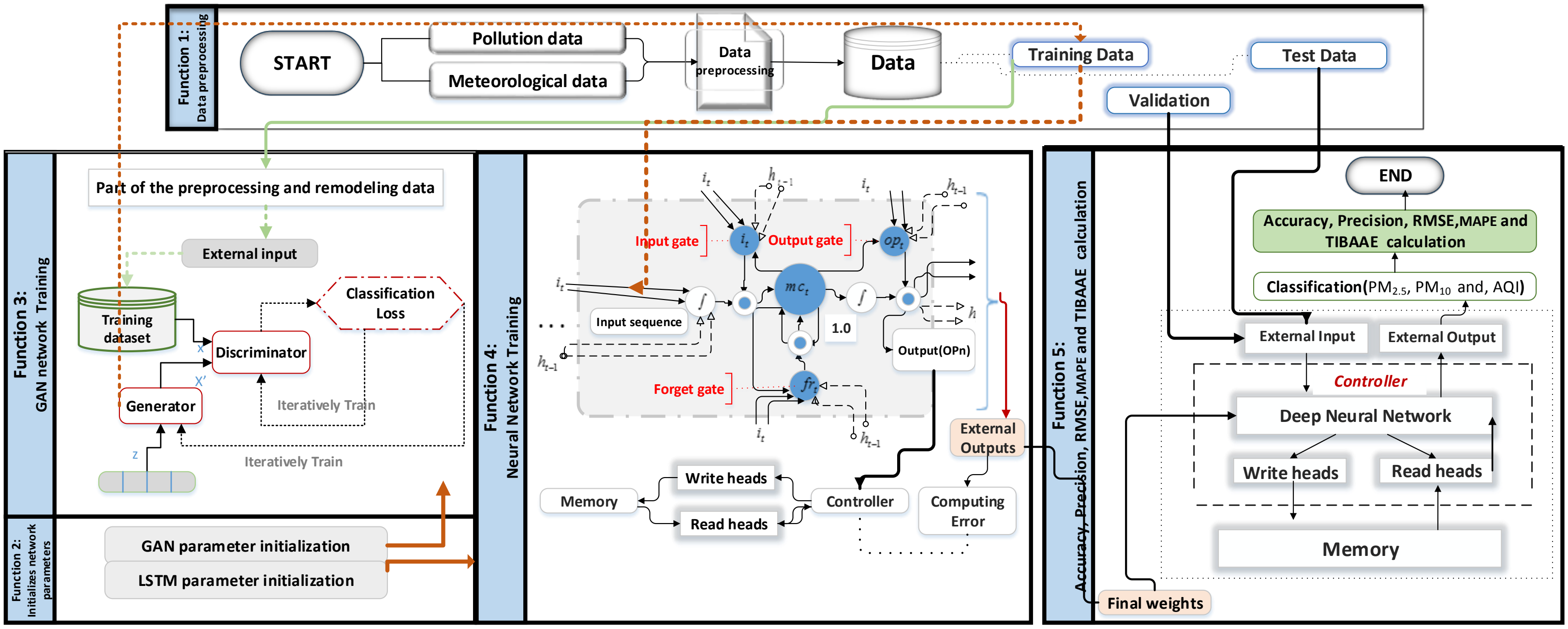


Figure 3: Applying deep learning and neural Turing machine along with GAN network (DAerosol. GAN. NTM:) in the aerosol prediction framework

3.1 Function 1: Data preprocessing

Figure 4 encompasses the data preprocessing steps in Function 1. After understanding the environment and the data, the data cleaning phase replaces the missing information with the correct ones. The noise data are also smoothed based on neighbors, and outlier data is then detected and removed from the data. Next, the scattered data from different sources are combined. The data conversion step uses normalization techniques to transfer values to a specific range; Equation 1 maps values to a particular range. This study also applies technical experts to supervise the selection of high-priority features and their impact on meteorological parameters in air pollution. Validation is checked using correlation and sensitivity analysis techniques. Finally, the data is compacted using the averaging compression technique to get processed. Section 4.1 describes the data preprocessing in more detail, as well.

$$X_n = \left({}^{(0.9(X - X_{\min}))} \big/ {}_{(X_{\max} - X_{\min})} \right) + 0.1 \qquad \text{Equation 1}$$

```
Function 1: Data preprocessing
Input = Pollution data & Meteorological dataset
Output = Dataset
Description: The combination of two Pollution data && Meteorological data to achieve a single dataset
    1.  Data cleaning for each one of the datasets (Pollution data & Meteorological data)
    2.  Data integration (The combination of two pollution data & Meteorological data)
    3.  Data transformation (using Normalization techniques) with Eq.12
    4.  Selection of compelling features (using the correlation analysis technique)
    5.  Data dimension reduction (using compression techniques)
```

Figure 4: Dataset preprocessing using DAerosol.NTM and DAerosol.GAN.NTM methods

3.2 Function 2-1: Adjusting the GAN parameters

Figure 5 shows the initialization function that sets the initial values of the network. Some values, such as the weights of neurons, are randomly assigned. It consists of two parts, Generator and Discriminator neural networks. The number of neurons and layers was evaluated by trial and error, and the best test results were taken as the output. In addition, the number of neurons in the input layer equals the length of the input. The problem is that there are no output variables, and the goal is to find interesting patterns in the data. Such uncorrected models are commonly referred to as the Unsupervised Form of Learning.

```
Function 2-1: Initializing the GAN network parameters
Input = number of layers, number of input nodes, number of the hidden node
Output = Topology of GAN (set the heads)
    1   Number of Layers = L
    2   Number of input nodes = length of sequence vector
    3   Number of hidden nodes = H
```

Figure 5: GAN neural network preparation pseudo-code for DAerosol.GAN.NTM method

3.3 Function 2-2: Initializing LSTM Network parameters

Figure 6 shows the initialization function that sets the initial values of the network. The weight of neurons is randomly assigned. Others, such as the number of neurons and layers, are evaluated by trial and error. In addition, the network's input and output layer neurons must be proportional to the length of the input data and the number of classes with which the network is to categorize the data, respectively.

```
Function 2-2: Initialize LSTM network parameters
Input: number of layers, number of input nodes, number of hidden nodes, number of output node
Output: The LSTM topology
    1.  Number of Layers=L
    2.  Number of input nodes=Length of sequence vector
    3.  Number of hidden nodes=H
    4.  Number of output node=number of classes
```

Figure 6: The pseudo-code for the initialization of DAerosol.NTM and DAerosol.GAN.NTM methods

3.4 Function 3-1: The GAN network training

Figure 7 shows that it is taught in two phases: Generator and Discriminator. An error backpropagation algorithm and a discriminant network error (cost function) are applied in training. The generating sector seeks to generate data so that the discriminator section cannot identify them as fraudulent. The discriminator section aims to identify the original numbers from the fake numbers. The cost function of each generating and separating network depends on the parameters of the two networks. Both types of costs are defined based on the discriminant error. The generator seeks to maximize the separator error while the discriminator minimizes it. In other words, the cost optimization process in GAN is a competitive optimization strategy (game theory). Notably, each network can only adjust its parameters in the training process.

```
Function 3: Algorithm "GAN" network training
Input = part of (10%) the preprocessing data and their reconstruction
Output = Accuracy=50%
    Discriminator network training,
    Selecting some random educational data randomly (X),
    Creating several random noise vectors and producing several counterfeit samples ($G(z)=X^*$),
    Error calculation ($X, X^*$) to train discriminator network weights and minimize discriminator network errors (At this stage, the weights of the generating network, are constant),
    Generator network training (the discriminator's training ratio of 5 to 1 compared to the generator),
    Creating several random noise vectors and producing several counterfeit samples ($G(z)=X^*$),
    Error calculation ($x^*$) to train generator network weights and maximize discriminator network errors,
    The weights of the discriminator grid are considered constant,
    Achieving an optimal solution (Nash Equilibrium) in a GAN network that, at best, can detect 50/50.
```

Figure 7: The pseudo-code for GAN implementation

3.5 Function 4: Neural Network Training for DAerosol.NTM and DAerosol.GAN.NTM Methods

The purpose of this function is network training shown in Figure 8. Initially, this function receives the weight of the last step and sets up the network. It also receives input data as a vector and generates the correct output. It then compares the network output with the desired result to calculate the network error according to Equation 2 and the mean absolute percentage error via Equation 3. The network weight is then updated accordingly.

```
Function 4: DAerosol.NTM && DAerosol.GAN.NTM algorithm for network training
Input = Dataset
Output = Accuracy
    Call Function 1 (data preprocessing)
    Initializing GAN network parameters
    Call Function 3 (GAN network training)
    "Training data" including both "actual data" and unsupervised data generated by the GAN network
    Initializing LSTM network parameters;
    Repeating the procedure (for each training input)
    Decision-making on forgetting gates with Equation 10;
    Training the LSTM network by using NTM external memory
    }
      If (the NTM external memory is not empty)
          Using read-head for reading previous weights from external memory (Equation 14, Equation 15)
      else
          Set weights of the network randomly.
    Decision-making for output gates using Equation 11;
    adjust the output of the neural network training function (value of weights, row of sequence vectors);
    Checking the error with Root-Mean-Square deviation and Mean Absolute Percentage Error via (Equation 2, Equation 3);
    Using write-head for writing best weights to external Memory with (Equation 16, Equation 17);
    }
    Ending loop
    return (Train accuracy; Train precision) with (Equation 4, Equation 5);
```

Figure 8: The pseudo-code of implementing GAN network combination with NTM with LSTM deep neural network controller

3.6 Function 5: Accuracy, Precision, RMSE, MAPE, and TIBAAE calculation

This section introduces essential parameters in the forecasting evaluation, including RMSE, MAPE, Accuracy, Precision, and the novel TIBAAE parameter. RMSE is the Root Mean Square Error (RMSE) calculated based on the known output and the machine output values during training and expressed by percentages according to Equation 2. MAPE is the Mean Absolute Percentage Error (MAPE), a measure of prediction accuracy of a forecasting method in statistics expressed by percentages according to Equation 3. It usually says that accuracy is a ratio defined as the percentage of correct predictions for the test data that can be calculated by dividing the number of accurate predictions by the total predictions according to Equation 4.

Precision is defined as the fraction of relevant examples among the whole predicted to be in a particular class. The machine output is compared with the known output to assess the precision of the network according to Equation 5.

$$RMSE = {}^{1}\!/_{n}((\sum_{i=1}^{lenghtofinput}(known.Output[i] - Machine.Output[i])^{2})^{1/2} \qquad \textbf{Equation 2}$$

$$\text{MAPE} = (({}^{1}\!/_{n}(\sum_{i=1}^{lenghtofinput} {}^{1}\!/_{(Known.Output[i])} \times (Known.Output[i] - Machine.Output[i])))\times 100) \qquad \textbf{Equation 3}$$

$$\text{Accuracy} = 1 - {}^{1}\!/_{n}((\sum_{i=1}^{lenghtofinput}(known.Output[i] - Machine.Output[i])^{2})^{1/2} \qquad \textbf{Equation 4}$$

$$\text{Precision} = 1 - (({}^{1}\!/_{n}(\sum_{i=1}^{lenghtofinput} {}^{1}\!/_{(Known.Output[i])} \times (Known.Output[i] - Machine.Output[i])))\times 100) \qquad \textbf{Equation 5}$$

Figure 9 introduces the TIBAAE parameter that describes how long before the aerosol event is; how predictable are the following days? For instance, by having the data of one day before the aerosol event by which accuracy the aerosol event happens in the next one/two/three, four, five, etc. days. Or by having the data from two days before the aerosol event by which accuracy the aerosol event happens in the following one/two/three/four/five, etc. days(Asaei-Moamam et al., 2023).

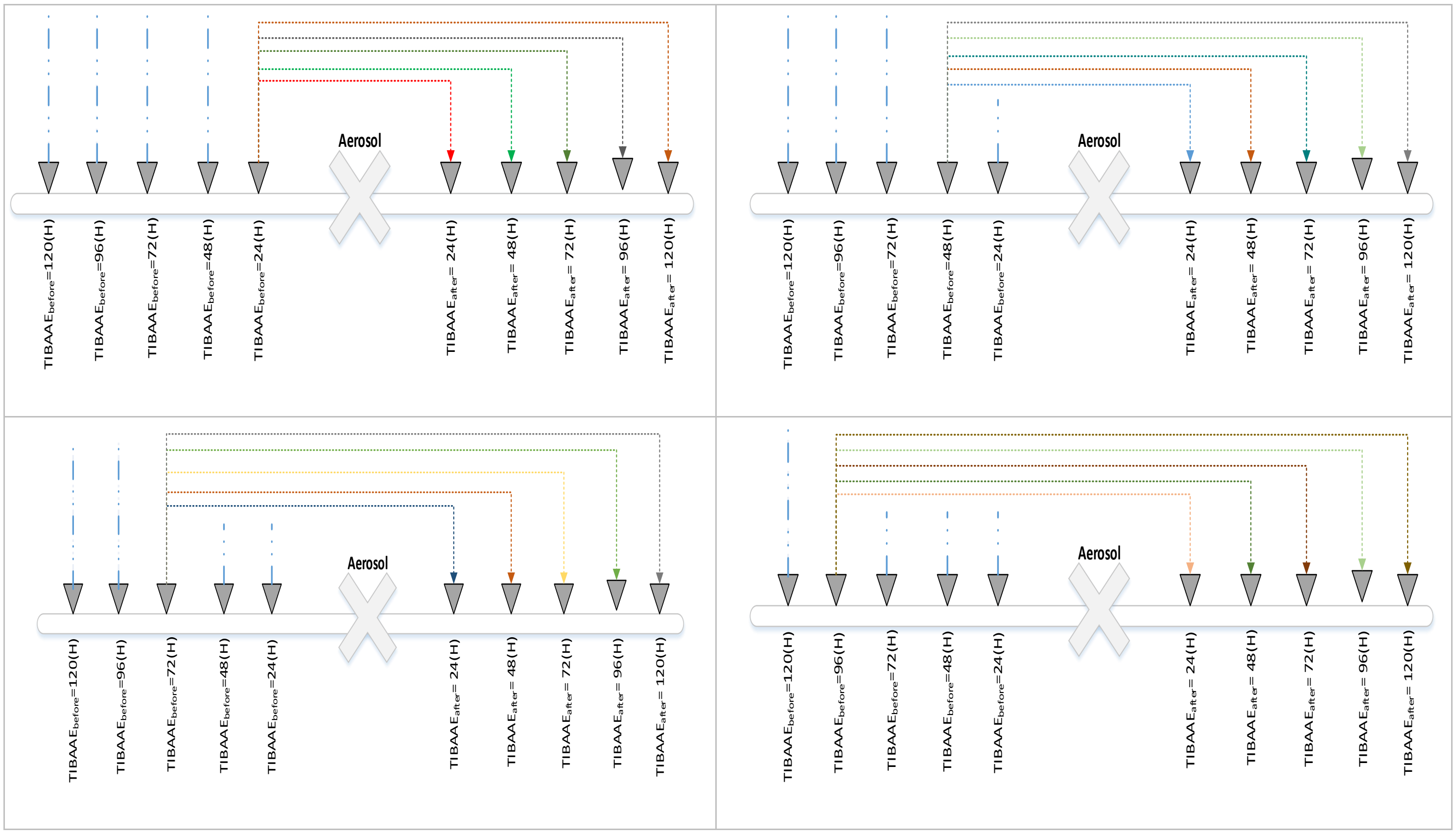


Figure 9: With data from one day to five days before the event, the possibility of fine dust occurring one day later, two days later, three days later, four days later, five days later

Figure 10 shows the approach to calculating the training, validation, and test prediction accuracy, precision, RMSE, MAPE, and TIBAAE.

```
Function 5: Validation and Test Evaluation
Input = Validation and Test dataset
Output = Classification accuracy, precision, RMSE, MAPE, TIBAAE;
    for each testing or validation input, do the following:
        set the testing or validation dataset as an input to the trained LSTM network;
        get LSTM network output;
    end for
```

```
5   Calculate accuracy, precision, RMSE, MAPE, and TIBAAE;
6   Return (accuracy, precision, RMSE, MAPE, TIBAAE);
```

Figure 10: Accuracy, Precision, RMSE, MAPE, and TIBAAE calculation pseudo-code used in DAerosol.NTM and DAerosol.GAN.NTM

3.7 The flow of information in DAerosol.GAN.NTM framework

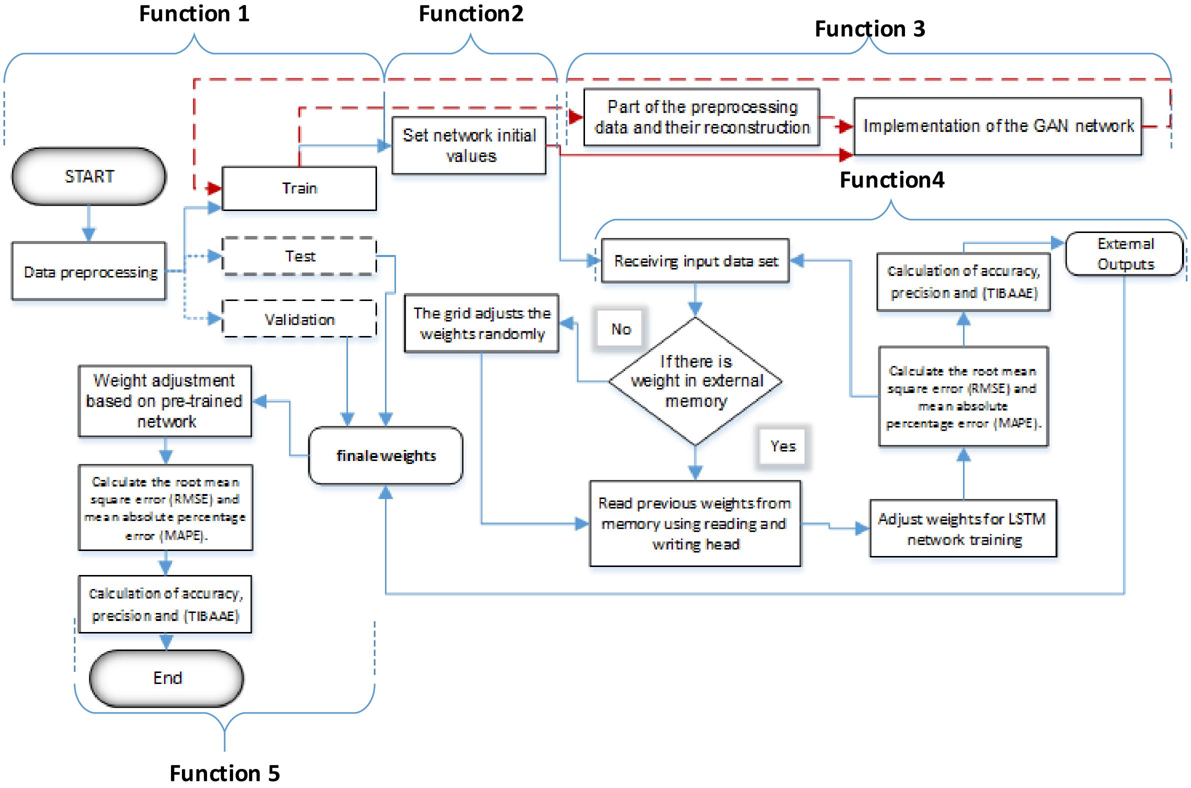

Function 1
Function2
Function 3
Part of the preprocessing data and their reconstruction
Implementation of the GAN network
Set network initial values
START
Train
Function4
Data preprocessing
Test
Validation
Receiving input data set
Calculation of accuracy, precision and (TIBAAE)
External Outputs
The grid adjusts the weights randomly
No
If there is weight in external memory
Calculate the root mean square error (RMSE) and mean absolute percentage error (MAPE).
Weight adjustment based on pre-trained network
finale weights
Yes
Calculate the root mean square error (RMSE) and mean absolute percentage error (MAPE).
Read previous weights from memory using reading and writing head
Adjust weights for LSTM network training
Calculation of accuracy, precision and (TIBAAE)
End
Function 5

Figure 11: DAerosol.GAN.NTM's Flowchart

Two DAerosol.NTM and DAerosol.GAN.NTM network training algorithms are stated in the above pseudo-code in Function 4 and Figure 8. The only part of distinct is DAerosol.GAN.NTM with the method DAerosol.NTM (lines 2 to 4, pseudo-code). A network with the same structure as the DAerosol.NTM algorithm has been used alongside the GAN network and the LSTM. In the DAerosol.GAN.NTM algorithm, the input data is first preprocessed and modified using Function 1 to fit the neural network (line 1, pseudo-code). After preparing the data, Function 2-1 (line 2, pseudo-code) must set the initial network GAN parameters. First, we assume that 10/100 of the data is missing or corrupt and enter the GAN network as input (line 3, pseudo-code). The training data, including actual and unsupervised data generated by the GAN network, are merged (line 4, pseudo-code). Then, to run the NTM network, the basic parameters of the LSTM network must be set using the 2-2 Function (line 5, pseudo-code) to train the network (Line 8 pseudo-code). According to the network structure, this algorithm reads the network's weights from the external memory by the read head and adjusts it in the network, but it should be checked whether there is a weight in the memory. The same random weights are used; otherwise, the weights are read from external memory and set in the grid (lines 10 to 13 pseudo-code). It is time to train the LSTM network by applying Function 4 (lines 14 and 15 pseudo-code) and calculating the network error (line 16 pseudo-code). Finally, the accuracy of the network operation is obtained using Function 5 (line 20 pseudo-code).

The following sections present a detailed description of Figure 11. The steps taken for controller network training are as follows:

- Step 1: The data preparation process
- Step 2: Adjusting the GAN network parameters
- Step 3: GAN network training according to Function 3
- Step 4: Preparing training data, including actual and unsupervised data generated by the GAN network.

- Step 5: Reading training data from the external input
- Step 6: Reading the previous weights stored in the NTM memory, provided that the external memory is not empty (Equation **14** and Equation **15**).

Note: This step does not occur in the LSTM deep neural network method without external memory.

- Step 7: Merging the previous network status, previously saved weights, and new input.

Input_{t+1}= [last weight from memory + new input data(signal)+last network state]

- Step 8: Updating the network weights
- Step 9: The network starts training with new data to generate the output via Equation 6.

$$Y = f(\sum_{i=1}^{N} W_i X_i)$$ Equation 6 -

The structure of network training is shown in Figure 12:

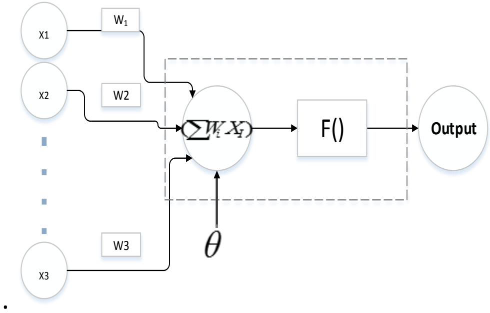


Figure 12: Network training structure

- Step 10: Calculating the error with Root-Mean-Square deviation and Mean Absolute Percentage Error via (Equation 2, Equation 3);
- Step 11: Store weights in external memory, as these weights are transferred to memory via the write head.

Note: This step does not occur in the LSTM deep neural network method without external memory.

- Step 12: Going to Step 6
- Step 13: Calculating the network output from the trained network for each test data
- Step 14: Calculating the accuracy through the network output and the desired (target) output

4 Experimental dataset, environment, and design

4.1 Dataset

The CRISP-DM data mining methodology (Azevedo & Santos, 2008) is used in data preprocessing steps. This methodology is a comprehensive method for data processing and divides the life cycle of data preprocessing into six phases, as shown in Figure 13(Asaei-Moamam et al., 2023).

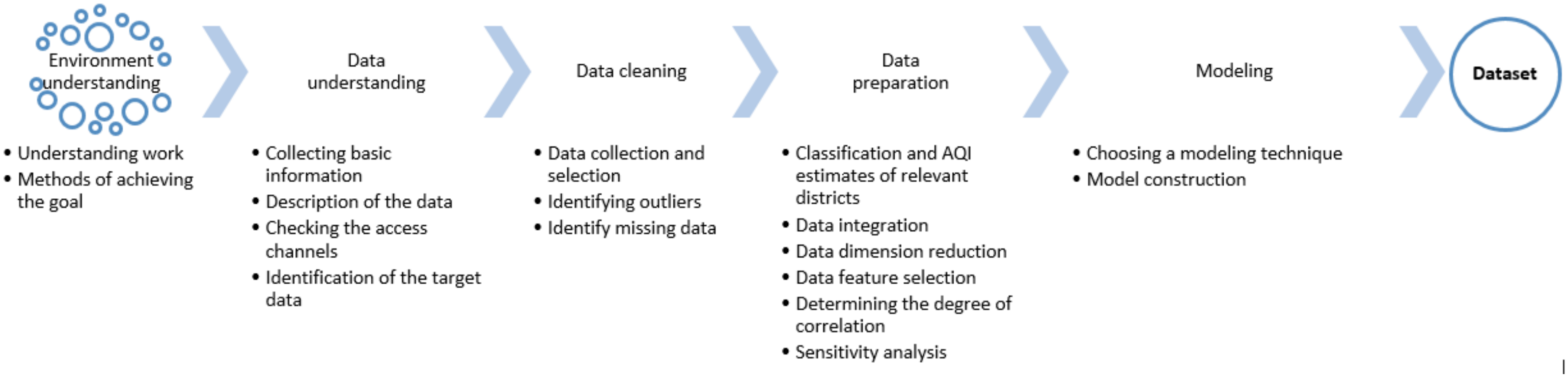

Environment understanding
Data understanding
Data cleaning
Data preparation
Modeling
Dataset
• Understanding work
• Methods of achieving the goal
• Collecting basic information
• Description of the data
• Checking the access channels
• Identification of the target data
• Data collection and selection
• Identifying outliers
• Identify missing data
• Classification and AQI estimates of relevant districts
• Data integration
• Data dimension reduction
• Data feature selection
• Determining the degree of correlation
• Sensitivity analysis
• Choosing a modeling technique
• Model construction

Figure 13: Data preprocessing steps

4.1.1 Environment understanding

Due to the lack of standard data on climate, pollution, and different climatic and geographical conditions, the present study only focuses on the parameters measured using the data from the City of Tehran in Iran. According to the 2018 estimate of the United Nations, Tehran is the 38th most populous city in the world, the most populous city in West Asia, the second most populated metropolis in the Middle East, and the most populous city and the capital of Iran(Castillo Esparcia & López Gómez, 2021)(Organización de las Naciones Unidas, 2018). According to the global air quality ranking, Tehran is usually among the ten most polluted cities in the world (Hosseini & Shahbazi, 2016). The capital Tehran is located on the southern slopes of the Alborz Mountain, between two mountain and desert valleys with about 730 square kilometers. Tehran Municipality has divided the city into 22 municipal districts and 122 urban districts to manage the city better. In the Ministry of Interior division, the city of Tehran is located in the center of Tehran province. It is neighboring Karaj and Shemiranat cities from the north, Damavand city from the east, Varamin, Rey, and Islamshahr cities from the south, and Shahryar and Karaj cities from the west (Nazmfar et al., 2019).

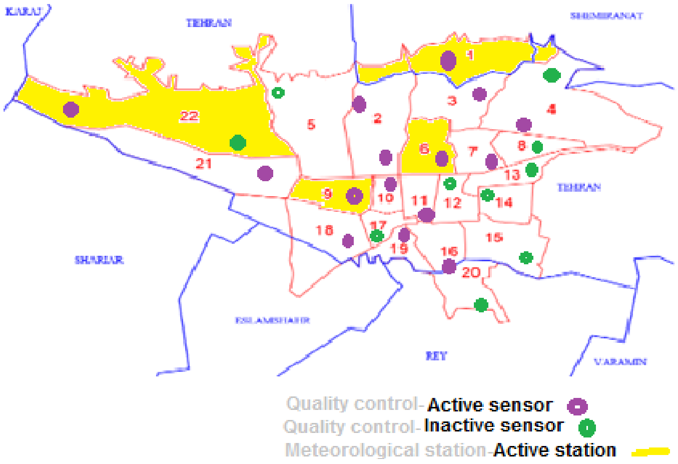

KARAJ
TEHRAN
SHEMIRANAT
1
22
5
2
3
4
21
6
7
8
13
TEHRAN
9
10
11
12
14
18
17
19
15
16
SHARIAR
20
ESLAMSHAHR
REY
VARAMIN
Quality control-Active sensor
Quality control-Inactive sensor
Meteorological station-Active station

Figure 14: Urban areas of Tehran (Nazmfar et al., 2019)

4.1.2 Data understanding: a partial view of the dataset

According to experts in the field of meteorology and air quality control, influential factors in predicting air pollution are sky conditions such as cover, type, layer, and height of clouds, horizontal visibility, pressure, wind direction and speed, relative humidity, temperature, carbon monoxide, ozone, nitrogen dioxide, and sulfur dioxide (Vallero, 2014). The data analysis indicated that $PM_{2.5}$ and PM_{10} have the highest AQI among other pollutants(Zhu et al., 2017). Table 2 shows a small portion of the dataset, and Table 3 shows the range of the collected parameters (Mohammadpour et al., 2018). Moreover, the data between (0 and 1) are normalized (Raturi & Prasad, 2018).

Table 2: A small sample of Datasets used in the experiments

	The lowest visible cloud	Horizontal view	Cloudiness	wind direction	wind speed	Temperature	Dew point	Sea level pressure	Station pressure	The amount of cloud is the lowest layer	Bottom layer cloud	Medium layer cloud	The amount of coverage of the first layer of the cloud	Type the first layer of the cloud	The base height of the first cloud layer	The amount of coverage of the second layer of the cloud	The second type of cloud layer	The base height of the second cloud layer	Relative humidity	Wet temperature	steam pressure	QNH pressure	Ozone	Carbon monoxide	Nitrogen dioxide	Sulfur dioxide	Solid particles with a diameter less than or equal to 10 mm	Solid particles with a diameter less than or equal to 2.5 mm	Air quality index
DATE	**h**	**vv**	**n**	**dd**	**ff**	**t**	**td**	**p**	**p0**	**nh**	**cl**	**cm**	**nl1**	**tl1**	**hl1**	**nl2**	**tl2**	**hl2**	**u**	**twet**	**ew**	**pqnh**	**O3**	**CO**	**NO2**	**SO2**	**PM 10**	**PM 2.5**	**AQI**
43589.25	0.45	0.46	0.1	0.12	0.1	0.1	0.1	0.15	0.3	0.1	0.1	0.1	0.1	0.11	0.44	0.1	0.1	0.38	0.12	0.12	0.13	0.94	1	1	1	1	0.54	0.71	0.49
43589.29	0.41	0.31	1	1	1	1	1	1	1	1	1	1	1	1	0.33	1	1	0.16	1	1	1	0.1	0.1	0.1	0.18	0.13	0.22	0.15	0.22
43589.33	0.41	0.31	1	1	1	1	1	1	1	1	1	1	1	1	0.33	1	1	0.16	1	1	1	0.1	0.11	0.1	0.18	0.1	0.19	0.14	0.19
43589.38	0.45	0.55	0.1	0.14	0.1	0.1	0.1	0.14	0.3	0.1	0.1	0.1	0.1	0.1	0.4	0.1	0.1	0.19	0.13	0.12	0.13	0.93	0.13	0.1	0.16	0.1	0.17	0.14	0.18
43589.42	0.41	0.31	1	1	1	1	1	1	1	1	1	1	1	1	0.33	1	1	0.16	1	1	1	0.1	0.15	0.1	0.14	0.1	0.17	0.13	0.155
43589.5	0.45	0.32	0.1	0.21	0.1	0.1	0.1	0.15	0.3	0.1	0.1	0.1	0.1	0.1	0.4	0.1	0.1	0.19	0.17	0.12	0.13	0.95	1	1	1	1	0.54	0.71	0.49
43589.63	0.45	0.55	0.1	0.12	0.1	0.1	0.1	0.15	0.3	0.1	0.1	0.1	0.1	0.1	0.4	0.1	0.1	0.19	0.15	0.12	0.13	0.94	1	1	1	1	0.54	0.71	0.49

Table 3: Range of air pollution variables

Parameter description	*Scale*	*Parameters*	*Min*	*Max*	*Mean*	*Standard deviation*
The lowest visible cloud	Meter	h	50	2500	1275	0.03998234875
Horizontal view	Degree	W	100	20000	10050	0.103439528
Cloudiness	Degree	n	2	3172	1587	0.409721572
Wind direction	Degree	dd	10	2428.631	1219.3155	0.378808213
Wind speed	Degree	ff	1	1943.608	972.304	0.409732491
Temperature	Degree	t	-9.8	1625.968	808.083892	0.402200747
Dew point	Degree	td	-17.7	1394.558	688.4290375	0.403780867
Sea level pressure	kg/cm2	p	989.1679	1346.518	1167.843169	0.385220176
Station pressure	Degree	p0	761.6	1293.615	1027.607462	0.325946986
The amount of cloud is the lowest layer	Degree	nh	761.6	1164.305	962.9526135	0.409606825
Bottom layer cloud	Degree	cl	1	1058.625	529.8126735	0.409200627
Medium layer cloud	Degree	cm	1	970.6095	485.8047451	0.408983303
The amount of coverage of the first layer of the cloud	Degree	nl1	1	895.9941	448.4970519	0.409823982
Type the first layer of the cloud	Km	tl1	1	832.4384	416.7191842	0.407124661
The base height of the first cloud layer	Meter	hl1	90	3000	1545	0.038858835
The amount of coverage of the second layer of the cloud	Millibar	nl2	1	794.466	397.7329872	0.408709075
The second type of cloud layer	Km	tl2	2	747.9518	374.9758996	0.408854975
The base height of the second cloud layer	Meter	hl2	240	9000	4620	0.090861262
Relative humidity	Degrees Celsius	u	5	790.8312	397.9156215	0.388401697
Wet temperature	Degree	twet	-10.2	751.9397	370.8698612	0.399058489
Steam pressure	Degree	ew	-18.85572	716.4769	348.8105916	0.395297729
QNH pressure	Degree	pqnh	730.1052	1034.4	882.2525861	0.385501341
Ozone	ppb	O3	1	660.1358	330.5679177	0.400348919
Carbon monoxide	ppm	CO	0.1	601.2626	300.6813031	0.408890217
Nitrogen dioxide	ppb	NO2	1	555.443	278.2215109	0.380254769
Sulfur dioxide	ppb	SO2	1	513.9647	257.4823508	0.404108716
Air quality index	Good, Average, Unhealthy for Sensitive Groups. Unhealthy, Very unhealthy, Hazardous	AQI	11	985	498	0.148421883
Solid particles with a diameter less than or equal to 10 mm	ug/m3	PM_{10}	1	985	493	0.170669391
Solid particles with a diameter less than or equal to 2.5 mm	ug/m3	$PM_{2.5}$	1	675	338	0.257377418

4.1.3 Data cleaning

The quality control center of Tehran collected the air pollution dataset by choosing 15 active sensors (from among 25 ones). Some sensors were removed due to the lack of data occasionally, failure, and maintenance of sensors. Finally, 15 air pollution sensors and four meteorological stations were investigated in different parts of Tehran. The suitable dataset ranges from December 30, 1999, to May 4, 2019; this dataset includes 882,735 air pollution records and 226,330 meteorological records (1,109,065 records). Four selected meteorological stations and fifteen active sensors of areas exposed to air pollution are in Table 4.

Table 4: Name of selected statistical samples

Row	Areas subject to air pollution	Meteorological stations
1	District 1 – Aqdasiyah	District 1 - Shemiranat

2	District 2 – Sharif	District 6 - Geophysics
3	District 2 – Shahrdari	District 9 - Mehrabad
4	District 3 – Doroos	District 22 - Chitgar
5	District 4 - Shahrdari	
6	District 6 - Tarbiat Modarres	
7	District 7 - Setade bohran	
8	District 9 - Shahrdari	
9	District 10 - Shahrdari	
10	District 11 - Shahrdari	
11	District 16 - Shahrdari	
12	District 18 - Shahrdari	
13	District 19 - Shahrdari	
14	District 21 - Shahrdari	
15	District 22 - Shahrdari	

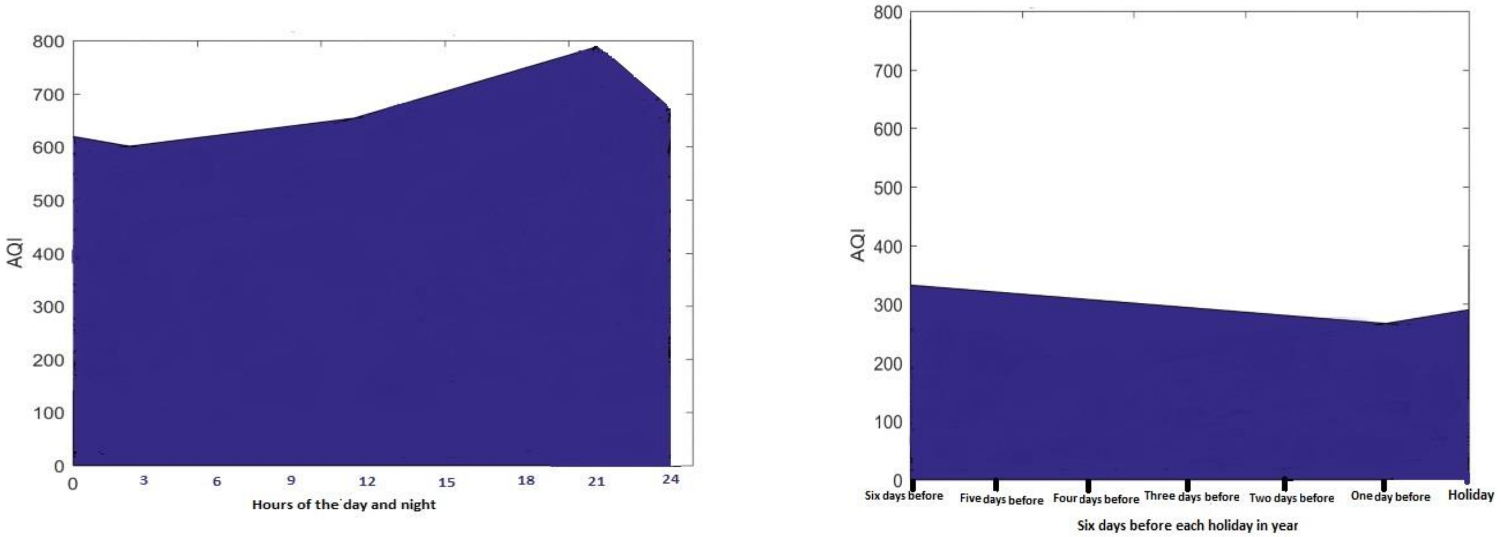
4.1.4 Data preparation

4.1.4.1 Classification and AQI estimates of relevant districts

This section examines the average AQI air quality index for all categories intended for the entire city of Tehran. The diagrams in Figure 15 show the preliminary analysis of the data and the correlations between the air quality index (AQI) and meteorological indicators for various times and periods. Finally, a deep learning model was applied with hourly data to predict air pollution in Tehran city. Since MATLAB software only applies to numbers easily, the data for each day is coded, as shown in Table 5.

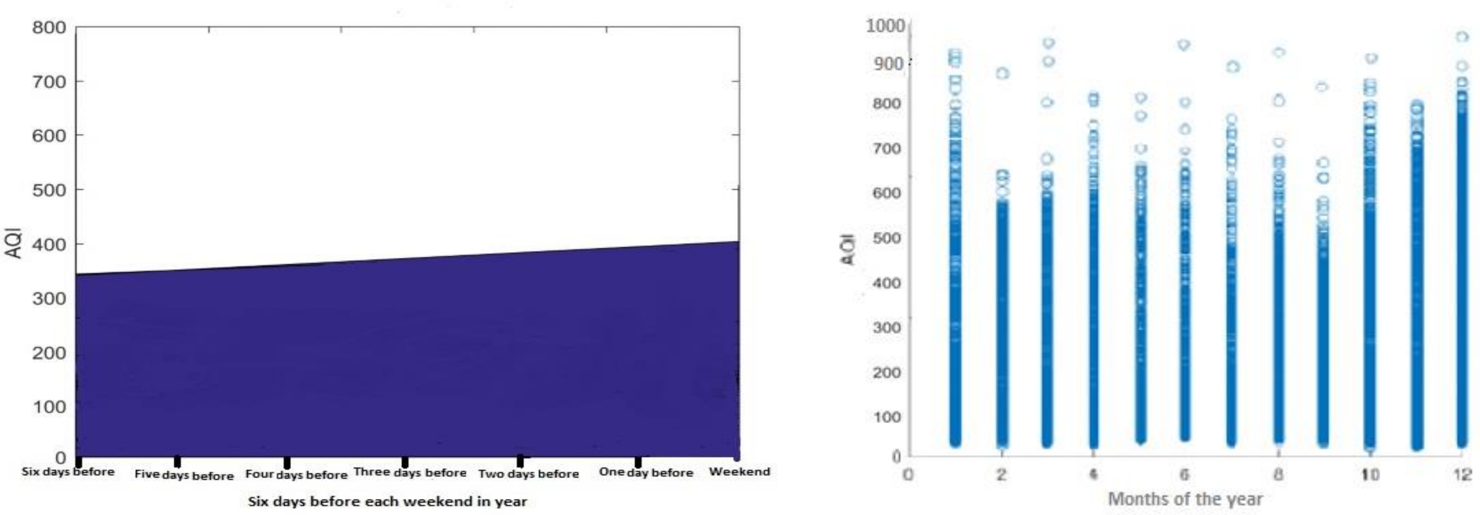
Table 5: Numbers considered for the days of the week in the data table

Days of the Week	Number
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6
Sunday	7



A: Average AQI for 24 hours of the day for twenty consecutive years

B: Average AQI for six days before each public holiday for twenty consecutive years



C: Average AQI for six days before each weekend for twenty consecutive years **D: Average AQI for each month of the year for twenty consecutive years**

Figure 15: AQI average air quality index of all studied categories in Tehran

Figure 15– A illustrates the average AQI for 24-hour days for twenty consecutive years, where twenty graphs were plotted for each day (140 charts in total). The x-axis indicates the hours of the day (0, 3, 6, 9, 12, 15, 18, 21, and 24), and the y-axis shows the average AQI index. As can be seen, the clock 21:00 is the most polluted time of the day, and there is the minimum pollution around the clock at 2:30 a.m. Pollution increases from noon onwards and reaches its peak at 9 p.m.

Figure 15– B illustrates all the official holidays and the average daily pollution until a week before the holiday. Assuming about 27 official holidays per year, 540 diagrams were drawn for 20 consecutive years (20 ×27 = 540). Figure 15-B summarizes all these diagrams. As can be seen, pollution starts declining six days before an official holiday but reaches its minimum value before the official holiday. However, before the official holidays, the pollution goes up again. In other words, the air population decreases just before the official holidays but increases in the final hours of the holidays.

Figure 15– C shows the average level of pollutants on the weekends during the year until a week before the holiday. Assuming 52 days for the weekends per year, 1040 diagrams were drawn for 20 consecutive years. These diagrams are summarized in Figure 15-C. As can be seen, pollution does not decrease on the weekends, but it increases with a relatively small slope.

Figure 15– D demonstrates the average monthly pollution for the whole year. There are 12 months in a year, then 20 points per month are specified for 20 consecutive years. The most polluted months are January, October, and December when air pollution increases mainly from early autumn to mid-winter. The changes in air pollution over the years are constant unless an extraordinary event happens.

4.1.4.2 **Data integration**

Tehran Meteorological Organization introduces four stations. The areas related to each station and the adjacent districts are categorized according to Figure 14, which are meteorological stations with quality control sensors. According to the World Meteorological Organization (WMO-1160), each meteorological station on a flat surface covers about 150 km (WMO, 2017). Thus, nearest neighbors are selected according to the geographic information system's latitude, longitude, and altitude (GIS).

Table 6 shows the category of areas in terms of air pollution in the meteorological stations: Shemiranat Station in District 1 (category 1), geophysics Station in District 6 (category 2), Mehrabad Station in District 9 (category 3), and Chitgar Station in district 22 (category 4). Air pollution data from those areas were also averaged to obtain unit data for each. It is worth noting that some areas were listed in several categories simultaneously due to their proximity to two or three meteorological stations. Each category with its associated districts is listed in the table below.

Table 6: Category of areas in terms of meteorological stations

	Name of stations			
	Shemiranat, **Category 1**	***Geophysics,*** **Category 2**	***Mehrabad,*** **Category 3**	***Chitgar,*** **Category 4**
1	District 1	District 2	District 9	District 21

2	District 3	District 3	District 10	District 22
3	District 4	District 6	District 16	-
4	-	District 7	District 18	-
5	-	District 8	District 19	-
6	-	District 10	District 21	-
7	-	District 11	District 2	-

4.1.4.3 Data dimension reduction

The air pollution data are generated hourly that must be synchronized with meteorological data produced every three hours. That is why this data must be changed from hourly to the average of every three hours. Then after integrating meteorological data with the relevant quality control data, the total dataset includes 58,921 (samples) with 82 (attributes). Then the subset of the dataset was selected from 2013 to 2020, including 10,000 records because this period had fewer sensor upgrades with minor data loss.

4.1.4.4 Data feature selection and determining the degree of correlation

With several technical experts' contributions, 29 features were selected from among 82 meteorological parameters affecting air pollution. The correlation coefficient is a statistical tool to determine the type and degree of relationship between one quantitative variable and another. The correlation coefficient is one of the criteria used to determine the correlation between two variables. The correlation coefficient shows the relationship's intensity and the type of relationship (direct or inverse). This coefficient is between -1 and 1 and equals zero if there is no relationship between the two variables. Table 7 shows the correlation coefficient of the independent and dependent variables varying. The closer the correlation coefficient is to one, the stronger the relationship between the variables. The closer the correlation coefficient is to zero, the weaker the relationship between the variables. A positive correlation coefficient indicates a direct relationship, while a negative one indicates an inverse relationship between them. These observations show a direct correlation between the Quality Control data and an indirect correlation between meteorological data.

Table 7: The degree of correlation between independent variables and dependent variables

Parameter description	Parameter	Correlation PM_{10}	Correlation $PM_{2.5}$	Correlation AQI
The lowest visible cloud	**h**	-0.626856278	-0.59053065178	-0.6697409903
Horizontal view	**w**	-0.607703276	-0.57248446361	-0.6492776859
Cloudiness	**n**	-0.623519478	-0.58738665652	-0.6661759105
Wind direction	**dd**	-0.558102351	-0.52575119044	-0.5962834462
Wind speed	**ff**	-0.537909738	-0.50672655177	-0.5747094481
Temperature	**t**	-0.527803925	-0.49720533264	-0.5639122357
Dew point	**td**	-0.253686353	-0.23896763935	-0.2710416335
Sea level pressure	**p**	-0.648549046	-0.61097019671	-0.6929178131
Station pressure	**p0**	-0.648429866	-0.61085789546	-0.6927904738
The amount of cloud is the lowest layer	**nh**	-0.564882591	-0.53213933215	-0.6035275445
Bottom layer cloud	**cl**	-0.580665947	-0.54701005421	-0.6203906691
Medium layer cloud	**cm**	-0.552014289	-0.5200152228	-0.5897788765
The amount of coverage of the first layer of the cloud	**nl1**	-0.565542989	-0.5327614680	-0.6042330393
Type the first layer of the cloud	**tl1**	-0.632773011	-0.5961056055	-0.6760625894
The base height of the first cloud layer	**hl1**	-0.639093125	-0.6020605133	-0.6828149926
The amount of coverage of the second layer of the cloud	**nl2**	-0.565368337	-0.5325969809	-0.6040465128
The second type of cloud layer	**tl2**	-0.590450388	-0.5562288318	-0.6308444865
The base height of the second cloud layer	**hl2**	-0.561194828	-0.5286648232	-0.599587485۴
Relative humidity	**u**	-0.562479899	-0.5298755763	-0.6009604690
Wet temperature	**twet**	-0.530023709	-0.4992966861	-0.5662838703
Steam pressure	**ew**	-0.588278379	-0.5541823374	-0.6285238308
QNH pressure	**pqnh**	-0.648566165	-0.6109863247	-0.6929361010
Ozone	**O3**	0.212247926	0.0937665708	0.2439902903
Carbon monoxide	**CO**	0.508057475	0.5595096392	0.5435975215
Nitrogen dioxide	**NO2**	0.535740491	0.6306802232	0.6179278878
Sulfur dioxide	**SO2**	0.465257284	0.5112756430	0.5090234275
Solid particles with a diameter less than or equal to 10 mm	**PM_{10}**	1	0.799675092	0.9799183616
Solid particles with a diameter less than or equal to 2.5 mm	**$PM_{2.5}$**		1	0.797171295
Air quality index	**AQI**			1

4.1.4.5 **Sensitivity analysis**

Sensitivity analysis means calculating and estimating the system's behavior; that is, how the system's output is sensitive to the values of the independent variables that are the input of that system. Table 8 calculates the ratio of input-to-output variables (impact factor) for both PM2.5 and PM10 particles.

To calculate the impact factor in this article, the following method has been implemented: 1) all input parameters are applied to the network, and after training the network, the obtained error is called E_k (network error for when all inputs are present), 2) By removing any of the input parameters in each run, the trained network is executed by applying other parameters. Actually, E_p is the network error when only input, p, was omitted, and 3) the equation $(E_p-E_k)/E_k$ calculates the impact factor that shows how the omitted input parameter impacts the network.

Table 9 and Table 10 present the sensitivity analysis for both PM2.5 and PM10. In the first step, the features were sorted from the maximum impact percentage to the minimum impact percentage in descending order, according to Table 8.

In the sensitivity analysis for PM2.5, features in descending order including PM10, NO2, pqnh, p, p0, hl1, tl1, h, n, Vv, ew, tl2, Co, cI, nI1, nI2, nh, u, hI2, dd, cm, SO2, ff, twet, t, td, O3, and for PM10, it includes pqnh, p, p0, hI1, tI1, h, n, vv, tI2, ew, cI, nI1, nl2, nh, u, hI2, dd, cm, ff, NO2, twet, t, CO, SO2, td, O3.

Then the sensitivity analysis was run to check the accuracy in terms of homogeneity. To this end, all independent variables were added to the shallow LSTM model simultaneously. Then, the variables with more minor effects were gradually removed from the equation one after another until obtaining the maximum value. The highlighted (Bold) parts indicate the features that have had the maximum impact on the model.

Table 8: Percentage of the effect of input variables on output

Input variable	**output**	**Impact factor**	**Input variable**	**output**	**Impact factor**
Solid particles with a diameter less than or equal to 10 mm	**PM_{10}**		**Solid particles with a diameter less than or equal to 2.5 mm**	**$PM_{2.5}$**	
QNH pressure	**pqnh**	0.648566165	Solid particles with a diameter less than or equal to 10 mm	**PM_{10}**	0.799671345
Sea level pressure	**p**	0.648549046	Nitrogen dioxide	**NO2**	0.630680223
Station pressure	**p0**	0.648429866	QNH pressure	**pqnh**	0.610986325
The base height of the first cloud layer	**hl1**	0.639093125	Sea level pressure	**p**	0.610970197
Type the first layer of the cloud	**tl1**	0.632773011	Station pressure	**p0**	0.610857895
The lowest visible cloud	**h**	0.626856278	The base height of the first cloud layer	**hl1**	0.602060513
Cloudiness	**n**	0.623519478	Type the first layer of the cloud	**tl1**	0.596105606
Horizontal view	**w**	0.607032760	The lowest visible cloud	**h**	0.590530652
The second type of cloud layer	**tl2**	0.590450388	Cloudiness	**n**	0.587386657
Steam pressure	**ew**	0.588278379	Horizontal view	**w**	0.572484464
Bottom layer cloud	**cl**	0.580665947	Carbon monoxide	**CO**	0.559509639
The amount of coverage of the first layer of the cloud	**nl1**	0.565542989	The second type of cloud layer	**tl2**	0.556228832
The amount of coverage of the second layer of the cloud	**nl2**	0.565368337	Steam pressure	**ew**	0.554182337
The amount of cloud is the lowest layer	**nh**	0.564882591	Bottom layer cloud	**cl**	0.547010054
Relative humidity	**u**	0.562479899	The amount of coverage of the first layer of the cloud	**nl1**	0.532761468
The base height of the second cloud layer	**hl2**	0.561194828	The amount of coverage of the second layer of the cloud	**nl2**	0.532596981
Wind direction	**dd**	0.558102351	The amount of cloud is the lowest layer	**nh**	0.532139332
Medium layer cloud	**cm**	0.552014289	Relative humidity	**u**	0.529875576
Wind speed	**ff**	0.537909738	The base height of the second cloud layer	**hl2**	0.528664823
Nitrogen dioxide	**NO2**	0.535740491	Wind direction	**dd**	0.52575119
Wet temperature	**twet**	0.530023709	Medium layer cloud	**cm**	0.520015223
Temperature	**t**	0.527803925	Sulfur dioxide	**SO2**	0.511275643
Carbon monoxide	**CO**	0.508057475	Wind speed	**ff**	0.506726552
Sulfur dioxide	**SO2**	0.465257284	Wet temperature	**twet**	0.499296686
Dew point	**td**	0.253686353	Temperature	**t**	0.497205333
Ozone	**O3**	0.212247926	Dew point	**td**	0.238967639
Solid particles with a diameter less than or equal to 2.5 mm	**PM2.5**	0	Ozone	**O3**	0.093766571
Air quality index	**AQI**	0	Air quality index	**AQI**	0

Table 9: Sensitivity Analysis for $PM_{2.5}$

Scenarios	Ozone	Dew point	Temperature	Wet temperature	Wind speed	Sulfur dioxide	Medium layer cloud	Wind direction	The base height of The second cloud layer	Relative humidity	The amount of cloud is the lowest layer	The amount of coverage of he second layer of the cloud	The amount of coverage of The first layer of the cloud	Bottom layer cloud	Carbon monoxide	The second type of cloud layer	Steam pressure	Horizontal view	Cloudiness	The lowest visible cloud	Type the first layer of the cloud	The base height of The first cloud layer	Station pressure	Steam pressure	QNH pressure	Nitrogen dioxide	Particulate matter 10 microns	Results (PM2.5)
	O3	td	t	twet	ff	SO2	cm	dd	hl2	u	nh	nl2	nl1	cl	CO	tl2	ew	Vv	n	h	tl1	hl1	p0	p	pqnh	NO2	PM10	
1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.1163
2	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.2185
3	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.3516
4	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.8494
5	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.3617
6	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.8146
7	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.1261
8	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.8886
9	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.6961
10	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.5996
11	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.6094
12	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.3666
13	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.6537
14	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.5927
15	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	0.3773
16	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	0.7586
17	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	0.0955
18	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	0.7194
19	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	0.1596
20	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	0.6598
21	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	0.9771
22	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	0.4898
23	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	0.8181
24	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	0.7973
25	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	0.8829
26	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	0.8444
27	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	0.7778

Table 10: Sensitivity Analysis for PM_{10}

Scenarios	Ozone	Dew point	Sulfur dioxide	Carbon monoxide	Temperature	Wet temperature	Nitrogen dioxide	Wind speed	Medium layer cloud	Wind direction	The base height of The second cloud layer	Relative humidity	The amount of cloud is The lowest layer	The amount of coverage of the second layer of the cloud	The amount of coverage of the first layer of the cloud	Bottom layer cloud	Steam pressure	The second type of cloud layer	Horizontal view	Cloudiness	The lowest visible cloud	Type the first layer of the cloud	The base height of the first cloud layer	Station pressure	Steam pressure	QNH pressure	Results (PM10)
	O3	td	SO2	CO	t	twet	NO2	ff	cm	dd	hl2	u	nh	nl2	nl1	cl	ew	tl2	vv	n	h	tl1	hl1	p0	p	pqnh	
1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.7648
2	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.7593
3	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.9232
4	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.2580
5	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.9448
6	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.6445
7	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.5125
8	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.3420
9	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.0001
10	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.9656

11	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.1302
12	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.3664
13	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	√	0.9469
14	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	√	0.7841
15	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	√	0.3938
16	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	√	0.7538
17	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	√	0.2453
18	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	√	0.9437
19	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	√	0.3188
20	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	√	0.7841
21	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	√	0.3008
22	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	√	0.5483
23	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	√	0.2618
24	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	√	0.2411
25	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	√	0.0026
26	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	√	0.2936

4.2 Experimental environment

Figure 16 illustrates the data preprocessing phase with Matlab R2018a software (P. Kim, 2017)and the experimental environment implemented in the Anaconda environment under the Python coding language 3.6 (Zocca et al., 2017). A desktop computer with 8GB of memory, Core7 CPU, and GeForce435m graphics was used for the implementation. The Keras library(Zocca et al., 2017), which uses the GPU-based Tensorflow framework (Brownlee, 2016), is a coding platform for Google Colab and Spyder(Vasilev et al., 2019), which implements deep neural networks. Some of the libraries required to implement NTM are as follows: 1) Theano (Brownlee, 2016) (Jason Brownlee, 2020) is a Python library that allows us to evaluate mathematical operations, including multi-dimensional arrays. 2) Lasagne (Jason Brownlee, 2020) is a lightweight library to build and train neural networks in Theano. 3) NumPy is Numerical Python, an open-source Python library used in almost every field of science and engineering. It should be noted that the only difference between this experimental environment and the usual deep neural networks is in using and implementing the neural Turing machine (NTM).

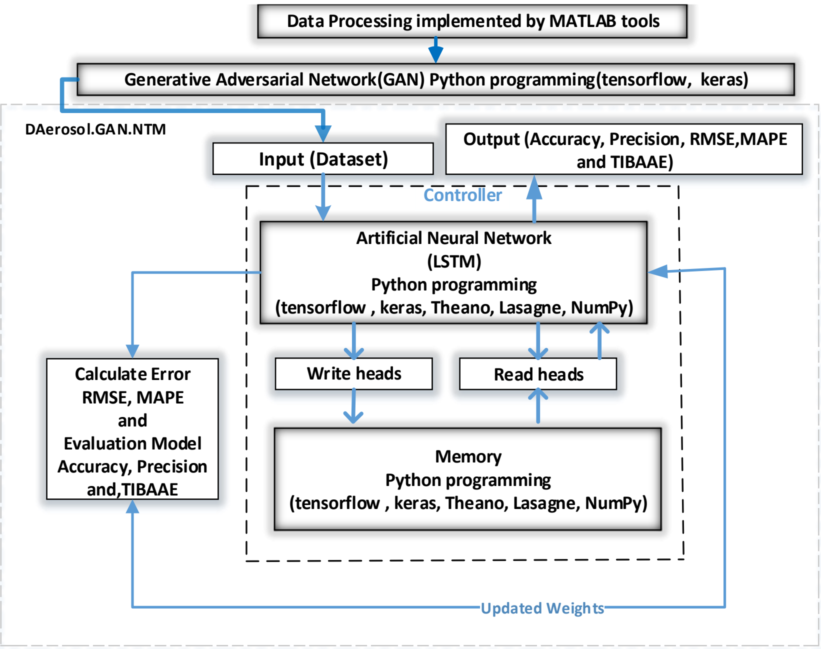

Data Processing implemented by MATLAB tools
Generative Adversarial Network(GAN) Python programming(tensorflow, keras)
DAerosol.GAN.NTM
Input (Dataset)
Output (Accuracy, Precision, RMSE,MAPE and TIBAAE)
Controller
Artificial Neural Network (LSTM) Python programming (tensorflow , keras, Theano, Lasagne, NumPy)
Calculate Error RMSE, MAPE and Evaluation Model Accuracy, Precision and,TIBAAE
Write heads
Read heads
Memory Python programming (tensorflow , keras, Theano, Lasagne, NumPy)
Updated Weights

Figure 16: The experimental environment

4.3 Experimental design and setup

This study uses the cross-validation method (Mohammadpour et al., 2018) to measure the model's efficiency. The training dataset is randomly divided into k folds (subfolders) of the same size, among which K-1 folds are considered the training and one-fold as the validation. Generally, 60% of the dataset is used for training, 20% for cross-validation to prevent over-fitting, and 20% for testing.

Table 11 and Table 12 show the experimental design and setup. Experiments are described as follows: Experiment-1 is for evaluating the DAerosol.NTM versus the baseline articles with 58,921 samples to evaluate prediction Accuracy, Precision, RMSE, and MAPE of $PM_{2.5}$, PM_{10}, and AQI parameters. Experiment-2 is for assessing the DAerosol.GAN.NTM versus the DAerosol.NTM with 10,000 samples to evaluate prediction Accuracy, Precision, RMSE, and MAPE of PM2.5, PM10, and AQI parameters. Experiment-3 considers DAerosol.GAN.NTM against DLSTM and DAerosol.NTM in terms of the time interval before and after the aerosol event (TIBAAE).

Table 11: Experimental design

Experiment no.	Experiment description	Dataset	Evaluated parameters	Parameter initialization	Hypothesis Support
Experiment-1	Comparing **DAerosol.NTM** with baseline research articles [(S. Kim et al., 2019)(Xayasouk et al., 2020)(Sharma et al., 2018)(Pengfei et al., 2018b)]	The provided dataset (Training, Test, and validation) with 58,921 samples	Evaluating prediction accuracy, precision, RMSE, and MAPE for PM2.5, PM10, and AQI	Initial network values	Hypothesis 1
Experiment-2	Comparing **DAerosol.GAN.NTM** and DAerosol.NTM	The provided dataset (Training, Test, and validation) with 10,000 samples	Evaluating prediction accuracy, precision, RMSE, and MAPE for PM2.5, PM10, and AQI	Initial network values	Hypothesis 2
Experiment-3	Comparing the time interval before and after predicting the aerosol (TIBAAE) event for **DAerosol.GAN.NTM**, **DAerosol.NTM** and **DLSTM** [(S. Kim et al., 2019)(Xayasouk et al., 2020)(Sharma et al., 2018)(Pengfei et al., 2018b)]	The provided dataset (Training, Test, and validation) with 10,000 samples	Evaluating TIBAAE for PM2.5, PM10, and AQI	Using the parameters set in the previous experiments	

Table 12: Experimental setup

Method	Adjustment parameters
Neural Turing Machine (NTM)	Controller=feedforward, Memory size=128, Learning rate=10^-4, Iterations=4000, Error=0
Generative Adversarial Network (GAN)	Backpropagation Algorithm, Topology=Generator&&Discriminator Neural Networks
Long Short-Term Memory (LSTM) Neural Network	4-layer, Activation function= tanh, [20, 15 ,9 ,6] hidden neurons
MLP	Feedforward net, Activation function= ReLU, Topology=9-layer [45,45,45, …, 45]

4.4 Experiments

4.4.1 Experiment-1: Evaluation of the proposed DAerosol.NTM compared to the baseline articles in terms of PM2.5, PM10, and AQI

Due to the unavailability of the data used in the baseline articles (S. Kim et al., 2019)(Xayasouk et al., 2020)(Sharma et al., 2018)(Pengfei et al., 2018b), this current study runs the baseline methods on the provided dataset to evaluate $PM_{2.5}$, PM_{10}, and AQI described in Table 13, Table 14, and Table 15.

The bar charts in Figure 17, Figure 18, and Figure 19 illustrate Table 13, Table 14, and Table 15 and support how the proposed model successfully predicts $PM_{2.5}$, PM_{10}, and AQI in the four target areas in Tehran in comparison to the baseline models.

Table 13: Comparison of DAerosol.NTM and baseline articles in terms of $PM_{2.5}$

Obtained results	Method	Train /Accuracy	Train /Precision	Train /RMSE	Validation /Accuracy	Validation /Precision	Validation /RMSE	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE
$PM_{2.5}$ (Pengfei et al., 2018)	**MLP**	0.80	0.59	0.20	0.79	0.48	0.21	0.72	0.45	0.28	0.55
$PM_{2.5}$ (Pengfei et al., 2018)	**DNN (MLP)**	0.87	0.85	0.13	0.82	0.50	0.18	0.74	0.49	0.26	0.51
$PM_{2.5}$ (Xayasouk et al., 2020)	**LSTM**	0.85	0.69	0.15	0.84	0.50	0.16	0.73	0.48	0.27	0.52
$PM_{2.5}$ (Xayasouk et al., 2020)	**DLSTM**	0.87	0.71	0.13	0.86	0.56	0.14	0.80	0.55	0.20	0.45
$PM_{2.5}$ (This research)	**DAerosol.NTM**	0.97	0.90	0.03	0.96	0.88	0.03	0.95	0.86	0.05	0.14

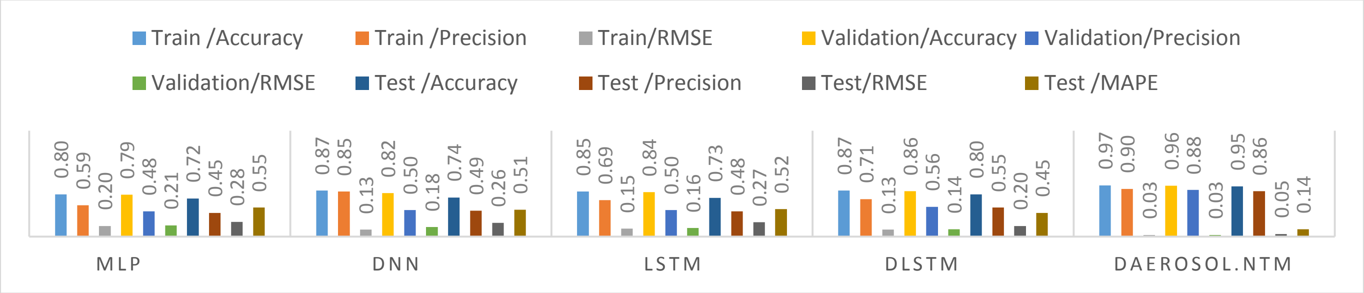


Figure 17: Comparison bar chart of DAerosol.NTM and baseline articles in terms of PM$_{2.5}$

Table 14: Comparison of DAerosol.NTM and baseline articles in terms of PM$_{10}$

Obtained results	Method	Train /Accuracy	Train /Precision	Train /RMSE	Validation /Accuracy	Validation /Precision	Validation /RMSE	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE
PM$_{10}$ (Xayasouk et al., 2020)	**LSTM**	0.86	0.70	0.14	0.84	0.51	0.16	0.74	0.49	0.26	0.51
	DLSTM	0.88	0.72	0.12	0.86	0.57	0.14	0.81	0.55	0.20	0.45
PM$_{10}$ (This research)	**DAerosol.NTM**	0.97	0.91	0.03	0.97	0.89	0.03	0.96	0.87	0.04	0.13

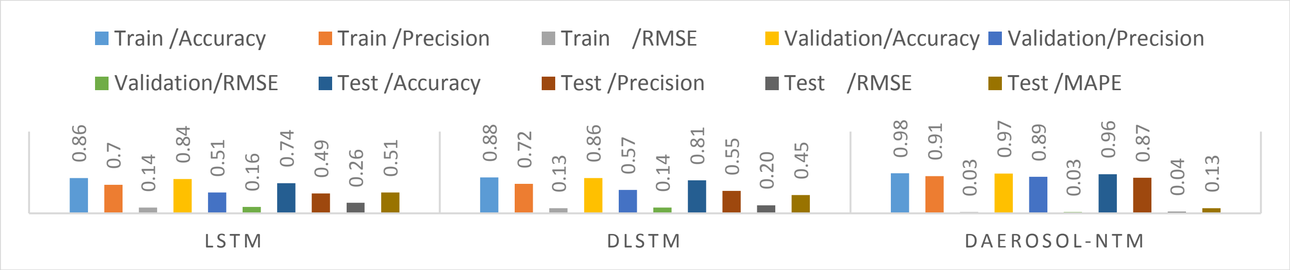


Figure 18: Comparison bar chart of DAerosol.NTM and baseline articles in terms of PM$_{10}$

Table 15: Comparison of DAerosol.NTM and baseline articles in terms of AQI

Obtained results	Method	Train /Accuracy	Train /Precision	Train /RMSE	Validation /Accuracy	Validation /Precision	Validation /RMSE	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE
AQI (Sharma et al., 2018)	**RNN**	0.79	0.57	0.21	0.77	0.46	0.23	0.70	0.44	0.30	0.56
	DLSTM	0.85	0.50	0.15	0.83	0.49	0.17	0.75	0.48	0.25	0.52
AQI (This research)	**DAerosol.NTM**	0.98	0.94	0.02	0.88	0.81	0.12	0.81	0.72	0.19	0.28

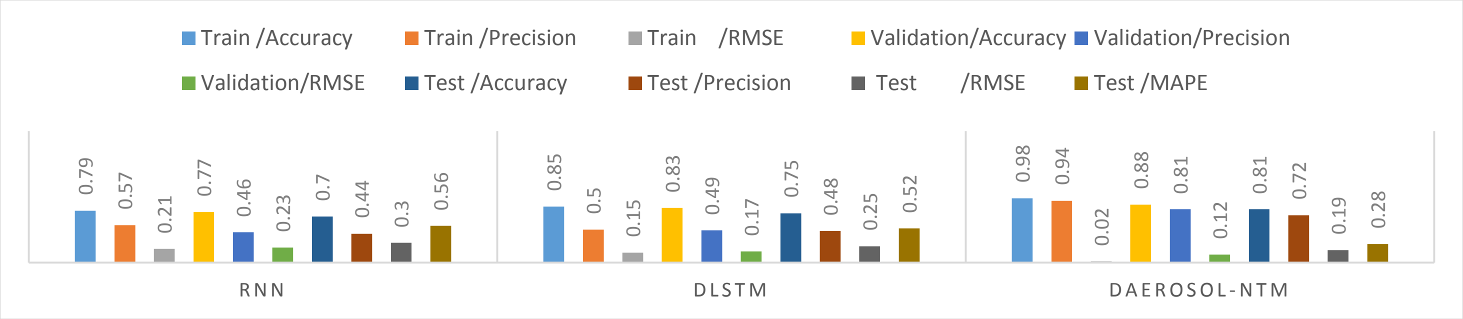


Figure 19: Comparison bar chart of DAerosol.NTM and baseline articles in terms of AQI

4.4.2 Experiment-2: Evaluating the proposed DAerosol.GAN.NTM compared to the DAerosol.NTM in terms of PM$_{2.5}$, PM$_{10}$, and AQI

Setting up GAN network parameters was time-consuming and performed on a trial-and-error basis. In the present study, the percentage of missing pollution data directly affecting the AQI estimates was 0.1 of the total training data. Finally, by performing this test, the prediction accuracy was increased, and the output was enhanced compared to the baseline method in Table 16 (DAerosol.NTM) and Table 17 (DAerosol.GAN.NTM). According to the various experiments performed on the data in this study, the number of data lost as a trial and error should not generally exceed 10/100 the actual data because the value of the real data will be paramount. The importance of this issue will be shown in the next step, which will generate unlabeled data for the coming days using the GAN network. It estimates the AQI and pollution occurrence time, which includes a minimum of 24 hours and a maximum of the next 120 hours.

Table 16: Prediction results of air pollution data in DAerosol.NTM method

Obtained results	Method	Train /Accuracy	Train /Precision	Train /RMSE	Validation /Accuracy	Validation /Precision	Validation /RMSE	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE
$PM_{2.5}$	DAerosol.NTM	0.97	0.89	0.03	0.94	0.85	0.04	0.95	0.86	0.03	0.14
PM_{10}		0.97	0.90	0.03	0.96	0.86	0.02	0.96	0.87	0.03	0.13
AQI		0.97	0.94	0.01	0.85	0.79	0.18	0.81	0.72	0.10	0.28

Table 17: Prediction results of air pollution data in DAerosol.GAN.NTM method

Obtained results	Method	Train /Accuracy	Train /Precision	Train /RMSE	Validation /Accuracy	Validation /Precision	Validation /RMSE	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE
$PM_{2.5}$	DAerosol.GAN.NTM	0.97	0.90	0.02	0.96	0.88	0.02	0.96	0.88	0.02	0.12
PM_{10}		0.98	0.91	0.02	0.97	0.89	0.02	0.97	0.89	0.02	0.11
AQI		0.98	0.95	0.01	0.97	0.93	0.02	0.97	0.94	0.02	0.06

4.4.3 Experiment-3: Comparing the time interval before and after predicting the aerosol event (TIBAAE) of DAerosol.GAN.NTM with DAerosol.NTM and DLSTM

Aerosol can be forecasted daily (24 hours) and hourly; however, hourly is usually not expected (Wu & Lin, 2019)(Jamal & Nodehi, 2017)(Hossain et al., 2020), and the General Directorate of Meteorology (Battan, 1979) mainly considers daily over hourly. This paper also applies daily forecasting to show the results in Table 18, Table 19, and Table 20. The time interval before predicting the aerosol event for the next twenty-four hours was obtained by considering data from the past five days to one day before the event. The bar chart in Figure 20 shows the aerosol concentration and future air quality control index ($TIBAAE_{after}$=24(H)), which was predicted with acceptable accuracy for four days before the Aerosol ($TIBAAE_{before}$=96(H)) occurrence using the DAerosol.NTM and DLSTM models.

Table 18: The next following one-day forecast of future $PM_{2.5}$ based on the hours before (24H to 120H) the Aerosol occurrence

Obtained results	Method	Train /Accuracy	Train /Precision	Train /RMSE	Validation /Accuracy	Validation /Precision	Validation /RMSE	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE	Prediction interval ($TIBAAE_{after}$= 24(H))
$PM_{2.5}$	DLSTM	0.83	0.52	0.17	0.81	0.52	0.19	0.79	0.49	0.21	0.51	$TIBAAE_{before}$=**120(H)**
		0.88	0.53	0.12	0.86	0.54	0.14	0.82	0.52	0.18	0.48	$TIBAAE_{before}$=**96(H)**
		0.81	0.50	0.19	0.80	0.50	0.20	0.76	0.47	0.24	0.53	$TIBAAE_{before}$=**72(H)**
		0.80	0.49	0.20	0.78	0.49	0.22	0.73	0.47	0.27	0.53	$TIBAAE_{before}$=**48(H)**
		0.77	0.47	0.23	0.77	0.48	0.23	0.71	0.46	0.29	0.54	$TIBAAE_{before}$=**24(H)**
	DAerosol.NTM	0.96	0.94	0.04	0.96	0.94	0.04	0.95	0.87	0.05	0.13	$TIBAAE_{before}$=**120(H)**
		0.97	0.95	0.03	0.97	0.95	0.03	0.96	0.90	0.04	0.10	$TIBAAE_{before}$=**96(H)**
		0.96	0.94	0.04	0.96	0.93	0.04	0.95	0.87	0.05	0.13	$TIBAAE_{before}$=**72(H)**
		0.95	0.92	0.05	0.95	0.93	0.05	0.93	0.85	0.07	0.15	$TIBAAE_{before}$=**48(H)**
		0.95	0.90	0.05	0.94	0.92	0.06	0.92	0.84	0.08	0.16	$TIBAAE_{before}$=**24(H)**

Table 19: The next following one-day forecast of future PM_{10} based on the hours before (24H to 120H) the Aerosol occurrence

Obtained results	Method	Train /Accuracy	Train /Precision	Train /RMSE	Validation /Accuracy	Validation /Precision	Validation /RMSE	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE	Prediction interval ($TIBAAE_{after}$= 24(H))
PM_{10}	DLSTM	0.85	0.55	0.15	0.84	0.54	0.16	0.79	0.51	0.21	0.49	$TIBAAE_{before}$=**120(H)**
		0.90	0.57	0.10	0.89	0.55	0.11	0.85	0.54	0.15	0.46	$TIBAAE_{before}$=**96(H)**
		0.86	0.55	0.14	0.85	0.53	0.15	0.79	0.52	0.21	0.48	$TIBAAE_{before}$=**72(H)**
		0.84	0.52	0.16	0.80	0.51	0.20	0.75	0.49	0.25	0.51	$TIBAAE_{before}$=**48(H)**
		0.80	0.51	0.20	0.79	0.50	0.21	0.73	0.47	0.27	0.53	$TIBAAE_{before}$=**24(H)**
	DAerosol.NTM	0.97	0.95	0.04	0.96	0.95	0.03	0.95	0.89	0.05	0.11	$TIBAAE_{before}$=**120(H)**
		0.98	0.96	0.02	0.98	0.96	0.02	0.96	0.90	0.04	0.10	$TIBAAE_{before}$=**96(H)**
		0.96	0.94	0.04	0.96	0.95	0.04	0.94	0.86	0.06	0.14	$TIBAAE_{before}$=**72(H)**
		0.95	0.93	0.05	0.95	0.94	0.05	0.93	0.85	0.07	0.15	$TIBAAE_{before}$=**48(H)**
		0.94	0.92	0.06	0.94	0.92	0.06	0.93	0.84	0.07	0.16	$TIBAAE_{before}$=**24(H)**

Table 20: The next following one-day forecast of future AQI quality control index based on the hours before (24H to 120H) the Aerosol occurrence

Obtained results	Method	Train /Accuracy	Train /Precision	Train /RMSE	Validation /Accuracy	Validation /Precision	Validation /RMSE	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE	Prediction interval (TIBAAE$_{after}$= 24(H))
AQI	**DLSTM**	0.84	0.53	0.16	0.82	0.53	0.18	0.80	0.50	0.20	0.50	TIBAAE$_{before}$=**120(H)**
		0.89	0.54	0.11	0.87	0.55	0.13	0.83	0.53	0.17	0.47	TIBAAE$_{before}$=**96(H)**
		0.83	0.52	0.17	0.81	0.52	0.19	0.78	0.49	0.22	0.51	TIBAAE$_{before}$=**72(H)**
		0.82	0.51	0.18	0.79	0.50	0.21	0.75	0.49	0.25	0.51	TIBAAE$_{before}$=**48(H)**
		0.79	0.50	0.21	0.78	0.50	0.22	0.73	0.48	0.27	0.52	TIBAAE$_{before}$=**24(H)**
	DAerosol.NTM	0.97	0.95	0.04	0.97	0.95	0.03	0.95	0.88	0.05	0.12	TIBAAE$_{before}$=**120(H)**
		0.98	0.96	0.02	0.98	0.96	0.02	0.96	0.90	0.04	0.10	TIBAAE$_{before}$=**96(H)**
		0.97	0.95	0.03	0.96	0.95	0.04	0.95	0.87	0.05	0.13	TIBAAE$_{before}$=**72(H)**
		0.95	0.93	0.05	0.95	0.93	0.05	0.94	0.86	0.06	0.14	TIBAAE$_{before}$=**48(H)**
		0.94	0.90	0.06	0.94	0.92	0.06	0.93	0.85	0.07	0.15	TIBAAE$_{before}$=**24(H)**

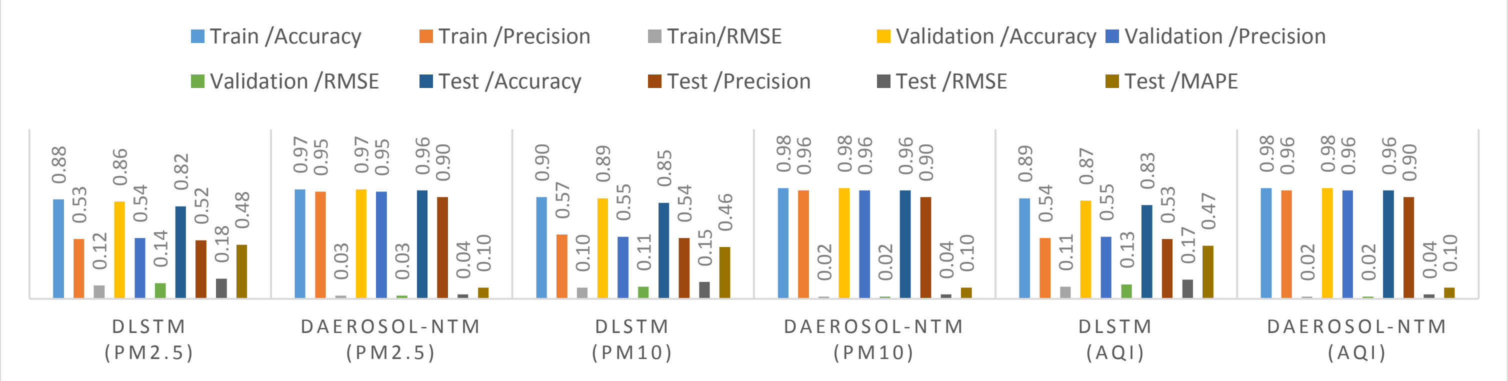


Figure 20: Comparison of TIBAAE$_{before}$=96(H) and TIBAAE$_{after}$=24(H) for DAerosol.NTM with DLSTM to the daily forecast of Aerosol concentration (PM$_{2.5}$, PM$_{10}$) and future quality control index (AQI) based

As AQI is the main focus of this research, the experiments will continue with AQI from now on. After generating the unsupervised data using the data of the one to five past days by the GAN network, all pollutants, including CO, SO2, O3, PM$_{2.5}$, PM$_{10}$, and NO2 in a specified period for at least three hours and at most the next 24 hours. The pollutants with the highest concentration ultimately determine the AQI value. The generated data is then entered into the NTM network as new data, which has already received the necessary training in predicting, and then evaluates the accuracy, precision, RMSE, and MAPE of classification. As shown in Table 21 (hourly data) and Table 22 (daily data), the concentration of the Future Air Quality Control Index has been predicted with more acceptable accuracy within four days before aerosol occurrence using the combined GAN and NTM network.

Table 21: Forecast results of minimum TIBAAE$_{after}$= 3(H) and maximum TIBAAE$_{after}$= 24(H) of air pollution data based on (TIBAAE$_{before}$= 24(H) to 120(H)) in the proposed framework.

Obtained results	Method	TEST /Accuracy	TEST /Precision	TEST /RSME	Test /MAPE	Time	Prediction interval (TIBAAE$_{after}$= 24(H))
AQI	**DAerosol.GAN.NTM**	0.95	0.93	0.04	0.07	3 AM	TIBAAE$_{before}$=**120(H)**
		0.94	0.93	0.04	0.07	6 AM	
		0.89	0.91	0.11	0.09	9 AM	
		0.88	0.91	0.12	0.09	12 Afternoon	
		0.87	0.91	0.13	0.09	15 PM	
		0.87	0.90	0.12	0.1	18 PM	
		0.85	0.90	0.14	0.1	21 PM	
		0.84	0.89	0.14	0.11	00 PM	
		0.97	0.95	0.02	0.05	3 AM	TIBAAE$_{before}$=**96(H)**
		0.96	0.94	0.03	0.06	6 AM	
		0.92	0.93	0.10	0.07	9 AM	
		0.89	0.92	0.11	0.08	12 Afternoon	
		0.89	0.92	0.11	0.08	15 PM	
		0.88	0.91	0.12	0.09	18 PM	
		0.86	0.90	0.14	0.1	21 PM	
		0.86	0.90	0.14	0.1	00 PM	

0.97	0.93	0.02	0.07	3 AM	TIBAAE$_{before}$=**72(H)**
0.96	0.93	0.03	0.07	6 AM	
0.87	0.91	0.12	0.09	9 AM	
0.87	0.90	0.12	0.1	12 Afternoon	
0.86	0.90	0.13	0.1	15 PM	
0.86	0.89	0.13	0.11	18 PM	
0.84	0.88	0.15	0.12	21 PM	
0.83	0.87	0.16	0.13	00 PM	
0.95	0.93	0.04	0.07	3 AM	TIBAAE$_{before}$=**48(H)**
0.94	0.92	0.05	0.08	6 AM	
0.85	0.89	0.14	0.11	9 AM	
0.84	0.89	0.15	0.11	12 Afternoon	
0.84	0.88	0.15	0.12	15 PM	
0.84	0.88	0.15	0.12	18 PM	
0.81	0.86	0.18	0.14	21 PM	
0.81	0.85	0.18	0.15	00 PM	
0.95	0.90	0.04	0.1	3 AM	TIBAAE$_{before}$=**24(H)**
0.94	0.89	0.05	0.11	6 AM	
0.81	0.85	0.18	0.15	9 AM	
0.81	0.85	0.18	0.15	12 Afternoon	
0.81	0.84	0.18	0.16	3 PM	
0.81	0.84	0.18	0.16	6 PM	
0.78	0.82	0.21	0.18	9 PM	
0.78	0.81	0.21	0.19	00 PM	

Table 22: The next following one-day forecast of future AQI quality control index based on the hours before (24H to 120H) the Aerosol occurrence

Obtained results	Method	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE	Prediction interval (TIBAAE$_{after}$= 24(H))
AQI	DAerosol.GAN.NTM	0.95	0.93	0.05	0.07	TIBAAE$_{before}$=120(H)
		0.97	0.95	0.03	0.05	TIBAAE$_{before}$=96(H)
		0.96	0.93	0.04	0.07	TIBAAE$_{before}$=72(H)
		0.95	0.93	0.05	0.07	TIBAAE$_{before}$=48(H)
		0.95	0.90	0.05	0.10	TIBAAE$_{before}$=24(H)

In continuation of the above Table 23, Table 24, Table 25, and Table 26 show the time interval before predicting the future quality control index (AQI) for the next up to 48 to 120 hours are obtained by considering data from the past five days to one day before. The results show that the closer we get from the twenty-four to the next 120 hours, the accuracy and precision decrease and the forecast error increases.

Table 23: The next following two-day forecast of future AQI quality control index based on the hours before (24H to 120H) the Aerosol occurrence

Obtained results	Method	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE	Prediction interval (TIBAAE$_{after}$= 48(H))
AQI	DAerosol.GAN.NTM	0.94	0.92	0.06	0.08	TIBAAE$_{before}$=120(H)
		0.96	0.94	0.04	0.06	TIBAAE$_{before}$=96(H)
		0.95	0.92	0.05	0.08	TIBAAE$_{before}$=72(H)
		0.94	0.92	0.06	0.08	TIBAAE$_{before}$=48(H)
		0.94	0.90	0.06	0.10	TIBAAE$_{before}$=24(H)

Table 24: The next following three-day forecast of future AQI quality control index based on the hours before (24H to 120H) the Aerosol occurrence

Obtained results	Method	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE	Prediction interval (TIBAAE$_{after}$= 72(H))
AQI	DAerosol.GAN.NTM	0.93	0.90	0.07	0.10	TIBAAE$_{before}$=120(H)
		0.94	0.92	0.06	0.08	TIBAAE$_{before}$=96(H)
		0.93	0.91	0.07	0.09	TIBAAE$_{before}$=72(H)
		0.92	0.90	0.08	0.10	TIBAAE$_{before}$=48(H)
		0.91	0.90	0.09	0.10	TIBAAE$_{before}$=24(H)

Table 25: The next following four-day forecast of future AQI quality control index based on the hours before (24H to 120H) the Aerosol occurrence

Obtained results	Method	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE	Prediction interval (TIBAAE$_{after}$= 96(H))
AQI	**DAerosol.GAN.NTM**	0.91	0.89	0.09	0.11	TIBAAE$_{before}$=120(H)
		0.93	0.91	0.07	0.09	TIBAAE$_{before}$=96(H)
		0.91	0.90	0.09	0.10	TIBAAE$_{before}$=72(H)
		0.90	0.90	0.10	0.10	TIBAAE$_{before}$=48(H)
		0.90	0.88	0.10	0.12	TIBAAE$_{before}$=24(H)

Table 26: The next following five-day forecast of future AQI quality control index based on the hours before (24H to 120H) the Aerosol occurrence

Obtained results	Method	Test /Accuracy	Test /Precision	Test /RMSE	Test /MAPE	Prediction interval (TIBAAE$_{after}$= 120(H))
AQI	**DAerosol.GAN.NTM**	0.90	0.87	0.10	0.13	**TIBAAE$_{before}$=120(H)**
		0.91	0.89	0.09	0.11	**TIBAAE$_{before}$=96(H)**
		0.89	0.88	0.11	0.12	**TIBAAE$_{before}$=72(H)**
		0.88	0.87	0.12	0.13	**TIBAAE$_{before}$=48(H)**
		0.86	0.87	0.14	0.13	**TIBAAE$_{before}$=24(H)**

4.5 Experimental discussion

The current study used the climate and pollution data of Tehran's capital city in Iran. The suggested DAerosol.GAN.NTM framework investigated the $PM_{2.5}$, PM_{10}, AQI, and TIBAAE parameters. Experiment-1 that supported the first hypothesis in Table 27, which could improve the prediction Accuracy/Precision/RMSE/MAPE of $PM_{2.5}$, PM_{10}, and the AQI quality control index due to considering the long history of meteorological information in the external memory of the deep learning neural network. The presented DAerosol.NTM framework, compared to the baseline articles (S. Kim et al., 2019)(Xayasouk et al., 2020)(Sharma et al., 2018)(Pengfei et al., 2018b), significantly enhanced accuracy by 14-31%, precision by 50-91%, Root Mean Square Error (RMSE) by 24-85%, and Mean Absolute Percentage Error (MAPE) by 46-75%.

Experiment-2 that supported the second hypothesis in Table 28, which is to improve the DAerosol.GAN.NTM method compared to DAerosol.NTM; the presented DAerosol.GAN.NTM framework, compared to the DAerosol.NTM, significantly enhanced accuracy by 1-19%, precision by 2-30%, (RMSE) by 33-80%, and (MAPE) by 14-78%.

Table 29, Experiment-3 supported the second hypothesis regarding the TIBAAE. Considering the data of **the previous four days (TIBAAE$_{before}$)**, the DAerosol.GAN.NTM could predict **the next twenty-four hours (TIBAAE$_{after}$)** with the best result. Compared to the DAerosol.NTM that previously reported the best results, DAerosol.GAN.NTM significantly increased TIBAAE (accuracy) by 1-18 %, TIBAAE (precision) by 5-33%, TIBAAE (RMSE) by 16-58 %, and TIBAAE (MAPE) by 33-68%.

Table 27: Summary of experimental result (Experiment-1) in support of Hypothesis-1

	Method																			
	DAerosol.NTM				**DLSTM**				**LSTM**				**DNN (MLP)**				**MLP**			
	Test/Accuracy	Test/Precision	Test /RMSE	Test /MAPE	Test/Accuracy	Test/Precision	Test /RMSE	Test /MAPE	Test/Accuracy	Test/Precision	Test /RMSE	Test /MAPE	Test/Accuracy	Test/Precision	Test /RMSE	Test /MAPE	Test/Accuracy	Test/Precision	Test /RMSE	Test /MAPE
$PM_{2.5}$	0.95	0.86	0.05	0.14	0.80	0.55	0.20	0.45	0.73	0.48	0.27	0.52	0.74	0.49	0.26	0.51	0.72	0.45	0.28	0.55
PM_{10}	0.96	0.87	0.04	0.13	0.81	0.55	0.20	0.45	0.74	0.49	0.26	0.51	0.75	0.51	0.25	0.49	0.73	0.46	0.28	0.54
AQI	0.81	0.72	0.19	0.28	0.75	0.48	0.25	0.52	0.70	0.46	0.30	0.54	0.71	0.47	0.29	0.53	0.69	0.44	0.31	0.56

Table 28: Summary of experimental result (Experiment-2) in support of Hypothesis−2

	Method							
	DAerosol.GAN.NTM				**DAerosol.NTM**			
	Test/Accuracy	**Test/Precision**	**Test /RMSE**	**Test /MAPE**	**Test/Accuracy**	**Test/Precision**	**Test /RMSE**	**Test /MAPE**

$PM_{2.5}$	0.96	0.88	0.02	0.12	0.95	0.86	0.03	0.14
PM_{10}	0.97	0.89	0.02	0.11	0.96	0.87	0.03	0.13
AQI	0.97	0.94	0.02	0.06	0.81	0.72	0.10	0.28

Table 29: Summary of experimental result (Experiment-3) in support of Hypothesis-2

			DAerosol.GAN. NTM					Average	DAerosol.NTM					Average	DAerosol.GAN. NTM					Average	DAerosol.NTM					Average	DAerosol.GAN. NTM					Average	DAerosol.NTM					Average	DAerosol.GAN. NTM					Average	DAerosol.NTM					Average
			Test/Accuracy						Test/Accuracy						Test/Precision						Test/Precision						Test/RMSE						Test/RMSE						Test/MAPE						Test/MAPE					
TIBAAE after			120(H)	96(H)	72(H)	48(H)	24(H)		120(H)	96(H)	72(H)	48(H)	24(H)		120(H)	96(H)	72(H)	48(H)	24(H)		120(H)	96(H)	72(H)	48(H)	24(H)		120(H)	96(H)	72(H)	48(H)	24(H)		120(H)	96(H)	72(H)	48(H)	24(H)		120(H)	96(H)	72(H)	48(H)	24(H)		120(H)	96(H)	72(H)	48(H)	24(H)	
PM2.5	TIBAAE before	120 (H)	0.89	0.91	0.93	0.94	0.96	0.93	0.77	0.80	0.87	0.91	0.95	0.86	0.89	0.91	0.90	0.93	0.94	0.91	0.70	0.74	0.79	0.82	0.87	0.78	0.11	0.09	0.07	0.06	0.04	0.07	0.23	0.20	0.13	0.09	0.05	0.14	0.11	0.09	0.10	0.07	0.06	0.09	0.30	0.26	0.21	0.18	0.13	0.22
		96 (H)	0.90	0.92	0.94	0.95	0.97	0.94	0.78	0.84	0.89	0.93	0.96	0.88	0.90	0.92	0.93	0.95	0.95	0.93	0.71	0.79	0.81	0.84	0.90	0.81	0.10	0.08	0.06	0.05	0.03	0.06	0.22	0.16	0.11	0.07	0.04	0.12	0.10	0.08	0.07	0.05	0.05	0.07	0.29	0.21	0.19	0.16	0.10	0.19
		72 (H)	0.88	0.90	0.93	0.94	0.96	0.92	0.75	0.81	0.85	0.90	0.95	0.85	0.88	0.90	0.91	0.92	0.93	0.91	0.68	0.74	0.78	0.83	0.87	0.78	0.12	0.10	0.07	0.06	0.04	0.08	0.25	0.19	0.15	0.10	0.05	0.15	0.12	0.10	0.09	0.08	0.07	0.09	0.32	0.26	0.22	0.17	0.13	0.22
		48 (H)	0.88	0.90	0.91	0.93	0.95	0.91	0.74	0.79	0.84	0.89	0.93	0.84	0.87	0.90	0.90	0.92	0.93	0.90	0.67	0.71	0.76	0.82	0.85	0.76	0.12	0.10	0.09	0.07	0.05	0.09	0.26	0.21	0.16	0.11	0.07	0.16	0.13	0.10	0.10	0.08	0.07	0.10	0.33	0.29	0.24	0.18	0.15	0.24
		24 (H)	0.86	0.90	0.90	0.93	0.95	0.91	0.73	0.78	0.82	0.88	0.92	0.83	0.87	0.88	0.90	0.90	0.90	0.89	0.65	0.70	0.75	0.78	0.84	0.74	0.14	0.10	0.10	0.07	0.05	0.09	0.27	0.22	0.18	0.12	0.08	0.17	0.13	0.12	0.10	0.10	0.10	0.11	0.35	0.30	0.25	0.22	0.16	0.26
PM10		120 (H)	0.89	0.90	0.92	0.93	0.96	0.92	0.77	0.81	0.86	0.91	0.95	0.86	0.88	0.90	0.90	0.93	0.94	0.91	0.70	0.73	0.78	0.82	0.89	0.78	0.11	0.10	0.08	0.07	0.04	0.08	0.23	0.19	0.14	0.09	0.05	0.14	0.12	0.10	0.10	0.07	0.06	0.09	0.30	0.27	0.22	0.18	0.11	0.22
		96 (H)	0.90	0.92	0.93	0.96	0.97	0.94	0.78	0.84	0.88	0.93	0.96	0.88	0.90	0.91	0.92	0.94	0.95	0.92	0.71	0.77	0.80	0.84	0.90	0.80	0.10	0.08	0.07	0.04	0.03	0.06	0.22	0.16	0.12	0.07	0.04	0.12	0.10	0.09	0.08	0.06	0.05	0.08	0.29	0.23	0.20	0.16	0.10	0.20
		72 (H)	0.88	0.90	0.93	0.94	0.96	0.92	0.76	0.82	0.86	0.91	0.94	0.86	0.88	0.90	0.91	0.92	0.93	0.91	0.67	0.73	0.78	0.81	0.86	0.77	0.12	0.10	0.07	0.06	0.04	0.08	0.24	0.18	0.14	0.09	0.06	0.14	0.12	0.10	0.09	0.08	0.07	0.09	0.33	0.27	0.22	0.19	0.14	0.23
		48 (H)	0.88	0.90	0.91	0.94	0.95	0.92	0.74	0.79	0.84	0.89	0.93	0.84	0.87	0.90	0.90	0.92	0.93	0.90	0.65	0.71	0.76	0.80	0.85	0.75	0.12	0.10	0.09	0.06	0.05	0.08	0.26	0.21	0.16	0.11	0.07	0.16	0.13	0.10	0.10	0.08	0.07	0.10	0.35	0.29	0.24	0.20	0.15	0.25
		24 (H)	0.86	0.90	0.91	0.93	0.95	0.91	0.74	0.79	0.83	0.89	0.93	0.84	0.87	0.88	0.90	0.90	0.90	0.89	0.65	0.70	0.75	0.78	0.84	0.74	0.14	0.10	0.09	0.07	0.05	0.09	0.26	0.21	0.17	0.11	0.07	0.16	0.13	0.12	0.10	0.10	0.10	0.11	0.35	0.30	0.25	0.22	0.16	0.26
AQI		120 (H)	0.90	0.91	0.93	0.94	0.95	0.93	0.77	0.81	0.86	0.91	0.93	0.86	0.87	0.89	0.90	0.92	0.93	0.90	0.70	0.74	0.79	0.83	0.88	0.79	0.10	0.09	0.07	0.06	0.05	0.07	0.23	0.19	0.14	0.09	0.07	0.14	0.13	0.11	0.10	0.08	0.07	0.10	0.30	0.26	0.21	0.17	0.12	0.21
		96 (H)	0.91	0.93	0.94	0.96	0.97	0.94	0.78	0.83	0.88	0.93	0.96	0.88	0.89	0.91	0.92	0.94	0.95	0.92	0.71	0.76	0.81	0.85	0.90	0.81	0.09	0.07	0.06	0.04	0.03	0.06	0.21	0.17	0.12	0.07	0.04	0.12	0.13	0.09	0.08	0.06	0.05	0.08	0.29	0.24	0.19	0.15	0.10	0.19
		72 (H)	0.89	0.91	0.93	0.95	0.96	0.93	0.75	0.81	0.85	0.90	0.95	0.85	0.88	0.90	0.91	0.92	0.93	0.91	0.68	0.73	0.78	0.82	0.87	0.78	0.11	0.09	0.07	0.05	0.04	0.07	0.25	0.19	0.15	0.10	0.05	0.15	0.12	0.10	0.09	0.08	0.07	0.09	0.32	0.27	0.22	0.18	0.13	0.22
		48 (H)	0.88	0.90	0.92	0.94	0.95	0.92	0.74	0.79	0.84	0.89	0.94	0.84	0.87	0.90	0.90	0.92	0.93	0.90	0.66	0.72	0.77	0.81	0.86	0.76	0.12	0.10	0.08	0.06	0.05	0.08	0.26	0.21	0.16	0.11	0.06	0.16	0.11	0.10	0.10	0.08	0.07	0.09	0.34	0.28	0.23	0.19	0.14	0.24
		24 (H)	0.86	0.90	0.91	0.94	0.95	0.91	0.73	0.78	0.82	0.88	0.93	0.83	0.87	0.88	0.90	0.90	0.90	0.89	0.65	0.70	0.75	0.79	0.85	0.75	0.14	0.10	0.09	0.06	0.05	0.09	0.27	0.22	0.18	0.12	0.07	0.17	0.13	0.12	0.10	0.10	0.10	0.11	0.35	0.30	0.25	0.21	0.15	0.25

5 Conclusions and future solutions

This study presented the **DAerosol.GAN.NTM** (Deep Aerosol – Generative Adversarial Networks - Neural Turing Machines), a GAN and NTM combined framework to accurately predict aerosol activity in terms of AQI values and the six principal underlying pollutants (SO2, NO2, $PM_{2.5}$, PM_{10}, O3, and CO). Compared to four baseline studies utilizing MLP, DNN, LSTM, and Deep LSTM (DLSTM)(S. Kim et al., 2019)(Xayasouk et al., 2020)(Sharma et al., 2018)(Pengfei et al., 2018b), the DAerosol.NTM significantly enhanced accuracy by 14-31%, precision by 50-91%, Root Mean Square Error (RMSE) by 24-85%, and Mean Absolute Percentage Error (MAPE) by 46-75%. Furthermore, compared to DAerosol.NTM, DAerosol.GAN.NTM improved accuracy by up to 19%, precision by up to 30%, RMSE by up to 80%, and MAPE by up to 78%. Finally, considering the data of **the previous four days ($TIBAAE_{before}$)**, the DAerosol.GAN.NTM could predict **the next twenty-four hours ($TIBAAE_{after}$)** with the best result. Compared to the DAerosol.NTM that previously reported the best results, DAerosol.GAN.NTM significantly increased TIBAAE (accuracy) by 1-18 %, TIBAAE (precision) by 5-33%, TIBAAE (RMSE) by 16-58 %, and TIBAAE (MAPE) by 33-68%.

In fact, the presented framework could use the NTM external storage features in the application of up to 20 years of data on pollutant and particulate matter, as well as twenty-two independent variables obtained from multiple meteorological stations in different districts in Tehran, Iran. This ability allowed DAerosol.GAN.NTM is the first to predict aerosol pollution and poor AQI values 24 to 120 hours before and after, outperforming DAerosol.NTM is the superior method in advanced air quality prediction.

Because unfavorable air quality environments could hinder socio-economic momentum significantly, some evaluated shortcomings of this study must be presented. Firstly, it focuses on the meteorological and geographical conditions of Tehran, Iran, primarily due to the lack of standardized data sources worldwide. Global partners and governmental bodies will execute data collection efforts and projects to implement similar models in other urban environments. Secondly that the raw data in previous baseline studies were unavailable, and no direct comparison could be carried out between the given methods and DAerosol.GAN.NTM within the exact geographic location. Lastly, the data retrieved from the Tehran meteorological and air quality stations were incomplete, or all together stations were unavailable, reducing the breadth of our study.

Several approaches can be implemented to enhance the DAerosol.GAN.NTM's accuracy and precision. A research study may apply NTM-derived machine learning models that modify memory architecture and produce neural computer models that DNCs can separate are proven stronger than NTM models. Combining GAN networks with DNC-based models can change memory architecture and allow the prediction of contamination parameters. The evolution of the control neural network using NEAT and PSO algorithms (or others of a similar function) can optimize NTM and DNC-based models. Lastly, studying DAerosol.NTM framework's time complexity and new aerosol-related tasks can enrich the NTM and DNC.

Despite the limitations, DAerosol.GAN.NTM performance was observed in the Tehran meteorological setting and could make it a vital tool in tackling the concerns of air quality prediction in real life. Different machine-learning approaches can be integrated with DAerosol.GAN.NTM to predict air quality indices accurately.

Practically, the framework must be trained with previous local data. After installing the sensors at desired locations, pollution indicators' live data are extracted, and then, the data are sent to the presented network as the test dataset. Subsequently, the proposed method predicts pollution accurately. The network will then be able to complete its training dataset with live data to make new predictions.

References

Ajibade, F. O., Adelodun, B., Lasisi, K. H., Fadare, O. O., Ajibade, T. F., Nwogwu, N. A., Sulaymon, I. D., Ugya, A. Y., Wang, H. C., & Wang, A. (2020). Environmental pollution and their socioeconomic impacts. In *Microbe Mediated Remediation of Environmental Contaminants* (pp. 321–354). Elsevier. https://doi.org/10.1016/B978-0-12-821199-1.00025-0

Akhtar, A., Masood, S., Gupta, C., & Masood, A. (2018). Prediction and analysis of pollution levels in Delhi using multilayer perceptron. *Advances in Intelligent Systems and Computing*, *542*, 563–572. https://doi.org/10.1007/978-981-10-3223-3_54

Appannagari, R. R. (2006). North Asian International Research Journal of ENVIRONMENTAL POLLUTION CAUSES AND CONSEQUENCES : A STUDY. *(Littler and Melanthiou, 2006)*, *3*(August 2017), 2454–9827.

Asaei-Moamam, Z.-S., Secondary, C. A., Author, C., Safi-esfahani, F., Mirjalili, S., Mohammadpour, R., & Term, L. (2023). DAerosol-NTM : Applying deep learning and neural Turing machine in aerosol prediction. *Neural Computing and Applications*, https://doi.org/10.1007/s00521-023-08868-4, [In press].

Azevedo, A., & Santos, M. F. (2008). KDD, semma and CRISP-DM: A parallel overview. *MCCSIS'08 - IADIS Multi Conference on Computer Science and Information Systems; Proceedings of Informatics 2008 and Data Mining 2008*, 182–185. https://recipp.ipp.pt/handle/10400.22/136

Baddeley, A. (1996). Working memory and executive control. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *351*(1346), 1397–1404. https://doi.org/10.1098/rstb.1996.0123

Battan, L. J. (1979). Fundamentals of meteorology. *Fundamentals of Meteorology*. https://doi.org/10.1007/978-3-030-52655-9

Boloukian, B., & Safi-Esfahani, F. (2020). Recognition of words from brain-generated signals of speech-impaired people: Application of autoencoders as a neural Turing machine controller in deep neural networks. *Neural Networks*, *121*, 186–207. https://doi.org/10.1016/j.neunet.2019.07.012

Brownlee, J. (2016). *Deep Learning With Python Develop Deep Learning Models On Theano And TensorFlow Using Keras i Deep Learning With Python*. https://books.google.com/books?hl=en&lr=&id=K-ipDwAAQBAJ&oi=fnd&pg=PP1&dq=related:CdZtlhleJl0J:scholar.google.com/&ots=oqUsULYjAt&sig=lRMn5nOLym7NKo0grv3zVuWX_F0#v=onepage&q&f=false

Castillo Esparcia, A., & López Gómez, S. (2021). Public opinion about climate change in United States, partisan view and media coverage of the 2019 United Nations climate change conference (COP 25) in Madrid. *Sustainability (Switzerland)*, *13*(7). https://doi.org/10.3390/su13073926

Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, *82*, 761–768. https://doi.org/10.1016/j.future.2017.08.043

Faradonbe, S. M., & Safi-Esfahani, F. (2020). A classifier task based on Neural Turing Machine and particle swarm algorithm. *Neurocomputing*, *396*, 133–152. https://doi.org/10.1016/j.neucom.2018.07.097

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2003). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, *3*(1), 115–143. https://doi.org/10.1162/153244303768966139

Graves, Alex and Mohamed, Abdel-rahman and Hinton, G. (2013). SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645--6649).

Graves, Alex and Wayne, Greg and Reynolds, Malcolm and Harley, Tim and Danihelka, Ivo and Grabska-Barwi{\'n}ska, Agnieszka and Colmenarejo, Sergio G{\'o}mez and Grefenstette, Edward and Ramalho, Tiago and Agapiou, J. and others. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471–476. https://doi.org/10.1038/nature20101

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing Machines. *ArXiv Preprint ArXiv:1410.5401*, 1–26. http://arxiv.org/abs/1410.5401

Greve, R. B., Jacobsen, E. J., & Risi, S. (2016). Evolving neural Turing machines for reward-based learning. *GECCO 2016 - Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, 117–124. https://doi.org/10.1145/2908812.2908930

Gulcehre, Caglar and Chandar, Sarath and Cho, Kyunghyun and Bengio, Y. (2016). Dynamic Neural Turing Machine with Soft and Hard Addressing Schemes. *ArXiv Preprint ArXiv:1607.00036*. http://arxiv.org/abs/1607.00036

Gulcehre, C., Chandar, S., Cho, K., & Bengio, Y. (2018). Dynamic neural Turing machine with continuous and discrete addressing schemes. *Neural Computation*, *30*(4), 857–884. https://doi.org/10.1162/NECO_a_01060

Han, Woo-jin and Cha, Sang-chang and Ha, H.-J. (2006). *Method and apparatus for multi-layered video encoding and decoding*.

Hochreiter, S., & Urgen Schmidhuber, J. (1997). Long Shortterm Memory. *Neural Computation*, *9*(8), 17351780. http://www7.informatik.tu-muenchen.de/~hochreit%0Ahttp://www.idsia.ch/~juergen

Hossain, Emam and Shariff, Mohd Arafath Uddin and Hossain, Mohammad Shahadat and Andersson, K. (2020). A Novel Deep Learning Approach to Predict Air Quality Index. In *Proceedings of International Conference on Trends in Computational and Cognitive Engineering* (pp. 367–381). https://doi.org/10.1007/978-981-33-4673-4_29

Hosseini, V., & Shahbazi, H. (2016). Urban Air Pollution in Iran. *Iranian Studies*, *49*(6), 1029–1046.

https://doi.org/10.1080/00210862.2016.1241587

Jamal, A., & Nodehi, R. N. (2017). A R T I C L E I N F O R M A T I O N PREDICTING AIR QUALITY INDEX BASED ON METEOROLOGICAL DATA: A COMPARISON OF REGRESSION ANALYSIS, ARTIFICIAL NEURAL NETWORKS AND DECISION TREE. In *Journal of Air Pollution and Health* (Vol. 2, Issue 1). http://japh.tums.ac.ir

Jason Brownlee. (2020). *Deep Learning With Python: Develop Deep Learning Models on Theano and ... - Jason Brownlee - Google Sách*. https://books.google.com.vn/books?hl=vi&lr=&id=K-ipDwAAQBAJ&oi=fnd&pg=PP1&dq=Deep+learning+with+Python:+develop+deep+learning+models+on+Theano+and+TensorFlow+using+Keras&ots=oqSp1H1iAv&sig=GljecgTwQ1aLP8uAeINCSSMzFNo&redir_esc=y#v=onepage&q=Deep learning with Python%3A develop deep learning models on Theano and TensorFlow using Keras&f=false

Jassim, M. S., & Coskuner, G. (2017). Assessment of spatial variations of particulate matter (PM10 and PM2.5) in Bahrain identified by air quality index (AQI). *Arabian Journal of Geosciences*, *10*(1), 1–14. https://doi.org/10.1007/s12517-016-2808-9

Kim, P. (2017). MATLAB Deep Learning. In *MATLAB Deep Learning*. Apress. https://doi.org/10.1007/978-1-4842-2845-6

Kim, S., Lee, J. M., Lee, J., & Seo, J. (2019). Deep-dust: Predicting concentrations of fine dust in Seoul using LSTM. *ArXiv Preprint ArXiv:1901.10106*, 8–10. http://arxiv.org/abs/1901.10106

Kök, I., Şimşek, M. U., & Özdemir, S. (2017). A deep learning model for air quality prediction in smart cities. *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, *2018-Janua*, 1983–1990. https://doi.org/10.1109/BigData.2017.8258144

Li, X., Peng, L., Hu, Y., Shao, J., & Chi, T. (2016). Deep learning architecture for air quality predictions. *Environmental Science and Pollution Research*, *23*(22), 22408–22417. https://doi.org/10.1007/s11356-016-7812-9

Liang, L., Wang, Z., & Li, J. (2019). The effect of urbanization on environmental pollution in rapidly developing urban agglomerations. *Journal of Cleaner Production*, *237*, 117649. https://doi.org/10.1016/j.jclepro.2019.117649

Lin, B., & Zhu, J. (2018). Changes in urban air quality during urbanization in China. *Journal of Cleaner Production*, *188*, 312–321. https://doi.org/10.1016/j.jclepro.2018.03.293

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. *ArXiv Preprint ArXiv:1506.00019*. http://arxiv.org/abs/1506.00019

Lovett, G. M., Tear, T. H., Evers, D. C., Findlay, S. E. G., Cosby, B. J., Dunscomb, J. K., Driscoll, C. T., & Weathers, K. C. (2009). Effects of Air Pollution on Ecosystems and Biological Diversity in the Eastern United States. *Annals of the New York Academy of Sciences*, *1162*(1), 99–135. https://doi.org/10.1111/j.1749-6632.2009.04153.x

Lu, J., Li, B., Li, H., & Al-Barakani, A. (2021). Expansion of city scale, traffic modes, traffic congestion, and air pollution. *Cities*, *108*, 102974. https://doi.org/10.1016/j.cities.2020.102974

Ma, Jun and Cheng, Jack CP and Lin, Changqing and Tan, Yi and Zhang, J. (2019). Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques. *Atmospheric Environment*, *214*, 116885. https://doi.org/10.1016/j.atmosenv.2019.116885

Malekmohammadi Faradonbeh, S., & Safi-Esfahani, F. (2019). A review on Neural Turing Machine. In *arXiv*. arXiv.

Mohammadi, M., Al-Fuqaha, A., Guizani, M., & Oh, J. S. (2018). Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services. *IEEE Internet of Things Journal*, *5*(2), 624–635. https://doi.org/10.1109/JIOT.2017.2712560

Mohammadpour, R., Asaie, Z., Shojaeian, M. R., & Sadeghzadeh, M. (2018). A hybrid of ANN and CLA to predict rainfall. *Arabian Journal of Geosciences*, *11*(18). https://doi.org/10.1007/s12517-018-3804-z

Nakata, M., Sano, I., & Mukai, S. (2015). Relation between aerosol characteristics and impact factor on climate and environment. *International Geoscience and Remote Sensing Symposium (IGARSS)*, *2015-November*, 2342–2345. https://doi.org/10.1109/IGARSS.2015.7326278

Nazmfar, H., Saredeh, A., Eshgi, A., & Feizizadeh, B. (2019). Vulnerability evaluation of urban buildings to various earthquake intensities: a case study of the municipal zone 9 of Tehran. *Human and Ecological Risk Assessment: An International Journal*, *25*(1–2), 455–474. https://doi.org/10.1080/10807039.2018.1556086

Organización de las Naciones Unidas. (2018). World Urbanization Prospects 2018. In *Department of Economic and Social Affairs. World Population Prospects 2018.* https://population.un.org/wup/

Pengfei, Y., Juanjuan, H., Xiaoming, L., & Kai, Z. (2018a). Industrial Air Pollution Prediction Using Deep Neural Network. *Communications in Computer and Information Science*, *951*, 173–185. https://doi.org/10.1007/978-981-13-2826-8_16

Pengfei, Y., Juanjuan, H., Xiaoming, L., & Kai, Z. (2018b). Industrial Air Pollution Prediction Using Deep Neural Network. *Communications in Computer and Information Science*, *951*, 173–185. https://doi.org/10.1007/978-981-13-2826-8_16

Qin, Y., Yin, Y., Wu, Z., & Shi, L. (2010). An observational study of atmospheric aerosol in the Shijiazhuang area. *2010 2nd IITA International Conference on Geoscience and Remote Sensing, IITA-GRS 2010*, *2*, 328–331. https://doi.org/10.1109/IITA-GRS.2010.5604099

Raturi, R., & Prasad, J. R. (2018). Recognition of Future Air Quality Index Using Artificial Neural Network. *International Research Journal of Engineering and Technology (IRJET)*, *5*, 2395--0056.

Rybarczyk, Y., & Zalakeviciute, R. (2018). Machine Learning Approaches for Outdoor Air Quality Modelling: A Systematic Review. *Applied Sciences*, *8*(12), 2570. https://doi.org/10.3390/app8122570

Sharma, Arjun and Mitra, Anirban and Sharma, Sumit and Roy, S. (2018). Estimation of Air Quality Index from Seasonal Trends Using Deep. *International Conference on Artificial Neural Networks*, 511–521. https://doi.org/10.1007/978-3-030-01424-7

Shams, R., & World.A. (2017). Assessing the accuracy of multiple regression model in forecasting air quality index (AQI) in Tehran. International Conference on New Research in Civil Engineering, Urban Management and Environment. https://civilica.com/doc/711061/

Sheng, M., Ma, Z., Jia, H., Mao, Q., & Dong, M. (2020). Face Aging with Conditional Generative Adversarial Network Guided by Ranking-CNN. *Proceedings - 3rd International Conference on Multimedia Information Processing and Retrieval, MIPR 2020*, 314–319. https://doi.org/10.1109/MIPR49039.2020.00071

Siegelmann, H. T., & Sontag, E. D. (1991). Turing computability with neural nets. *Applied Mathematics Letters*, *4*(6), 77–80. https://doi.org/10.1016/0893-9659(91)90080-F

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies %J Evolutionary Computation. In *Evolutionary Computation* (Vol. 10, Issue 2, pp. 99–127).

Stein, G., Gonzalez, A. J., & Barham, C. (2013). Machines that learn and teach seamlessly. *IEEE Transactions on Learning Technologies*, *6*(4), 389–402. https://doi.org/10.1109/TLT.2013.32

Turing, A. M. (1950). *M I N D A QUARTERLY REVIEW OF PSYCHOLOGY AND PHILOSOPHY I.-COMPUTING MACHINERY AND INTELLIGENCE*. Springer. https://academic.oup.com/mind/article/LIX/236/433/986238

Vallero, D. (2014). *Fundamentals of Air Pollution - Daniel Vallero - Google Books*. Academic Press. https://books.google.com/books?hl=en&lr=&id=iFcXAwAAQBAJ&oi=fnd&pg=PP1&dq=According+to+experts+in+the+field+of+meteorology+and+air+quality+control,+influential+factors+in+predicting+air+pollution+are+sky+conditions+such+as+cover,+type,+layer,+and+height+o

Vasilev, Ivan and Slater, Daniel and Spacagna, Gianmario and Roelants, Peter and Zocca, V. (2019). *Python Deep Learning: Exploring deep learning techniques and neural network ... - Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, Valentino Zocca - Google Books*. https://books.google.com/books?hl=en&lr=&id=ESKEDwAAQBAJ&oi=fnd&pg=PP1&dq=related:CdZtlhleJl0J:scholar.google.com/&ots=y4KwTVqf5T&sig=H8qDSdwZ_m27w8eVXh1ErtevJTE#v=onepage&q&f=false

Verbancsics, P., & Harguess, J. (2013). Generative NeuroEvolution for Deep Learning. *ArXiv Preprint ArXiv:1312.5355*. http://arxiv.org/abs/1312.5355

Wang, J., Zhang, X., Guo, Z., & Lu, H. (2017). Developing an early-warning system for air quality prediction and assessment of cities in China. *Expert Systems with Applications*, *84*, 102–116. https://doi.org/10.1016/j.eswa.2017.04.059

Weston, Jason and Bordes, Antoine and Chopra, Sumit and Rush, Alexander M and van Merri{\"e}nboer, Bart and Joulin, Armand and Mikolov, T. (2016, February 19). Towards AI-complete question answering: A set of prerequisite toy tasks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. https://arxiv.org/abs/1502.05698

WMO. (2017). Guide to the Global Observing System. In *Switzerland* (Issue 488). http://www.wmo.int/pages/prog/www/OSY/Manual/488_Guide_2007.pdf%0Ahttps://library.wmo.int/doc_num.php?explnum_id=4236

Wu, Q., & Lin, H. (2019). A novel optimal-hybrid model for daily air quality index prediction considering air pollutant factors. *Science of the Total Environment*, *683*, 808–821. https://doi.org/10.1016/j.scitotenv.2019.05.288

Xayasouk, T., Lee, H. M., & Lee, G. (2020). Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models. *Sustainability (Switzerland)*, *12*(6). https://doi.org/10.3390/su12062570

Yang, G., & Rush, A. M. (2019, February 27). Lie-access neural Turing machines. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. http://arxiv.org/abs/1602.08671

Yao, Kaisheng and Peng, Baolin and Zhang, Yu and Yu, Dong and Zweig, Geoffrey and Shi, Y. (2014). SPOKEN LANGUAGE UNDERSTANDING USING LONG SHORT-TERM MEMORY NEURAL NETWORKS. In *2014 IEEE Spoken Language Technology Workshop (SLT)* (pp. 189--194).

Zaremba, W., & Sutskever, I. (2015). Reinforcement Learning Neural Turing Machines - Revised. *ArXiv Preprint ArXiv:1505.00521*. http://arxiv.org/abs/1505.00521

Zangouei, H., & Asdaleh Fardi. (2017). Prediction of PM10 contamination in Mashhad using MLP artificial neural networks and Markov chain model. Journal of Applied Research in Geographical Sciences, vol. 17, no. 47, (pp. 39–59). https://iranjournals.nlai.ir/handle/123456789/578038

Zhao, J., & Peng, G. (2011). NEAT versus PSO for evolving autonomous multi-agents coordination on pursuit-evasion problem. *Lecture Notes in Electrical Engineering*, *133 LNEE*(VOL. 2), 711–717. https://doi.org/10.1007/978-3-642-25992-0_95

Zhu, S., Lian, X., Liu, H., Hu, J., Wang, Y., & Che, J. (2017). Daily air quality index forecasting with hybrid models: A case in China. *Environmental Pollution*, *231*, 1232–1244. https://doi.org/10.1016/j.envpol.2017.08.069

Zocca, Valentino and Spacagna, Gianmario and Slater, Daniel and Roelants, P. (2017). *Python Deep Learning - Valentino Zocca,* *Gianmario Spacagna, Daniel Slater, Peter Roelants - Google Books*. Packt Publishing Ltd. https://books.google.com/books?hl=en&lr=&id=2UEwDwAAQBAJ&oi=fnd&pg=PP1&dq=python+deep+learning%2Bbook&ots=RoCLFfYwei&sig=bjp_24blodyLrmkRNuVa0KdopRE#v=onepage&q=python deep learning%2Bbook&f=false

6 Appendix A - research concepts

This section addresses the theoretical and technical issues and specialized terms used in the study, including the AQI index, deep learning, deep short-term neural network, and the neural Turing machine.

6.1 AQI Index

AQI is an air quality indicator that reflects and evaluates the air quality status. Although the AQI scale is continuous, different descriptive categories have been implemented to ease public communication, as seen in Table 30(Diro & Chilamkurti, 2018)(Jassim & Coskuner, 2017).

One crucial factor to note about AQI is its calculation method. The concentrations are independently measured using six reported pollutant parameters (SO2, NO2, PM2.5, PM10, O3, and CO). Any pollutant's highest value is considered the AQI value(Zhu et al., 2017). For this reason, the calculation of AQI is driven by a single parameter, and the accurate prediction of all six pollutants is essential. In most urban areas, PM2.5 and PM10 are the leading drivers of AQI calculations and the dominating reason behind pollution and erosion.

Table 30: Different Levels (AQI) (Jassim & Coskuner, 2017)

Air Quality Index (AQI) Values	Levels of Health Concern	Colors
AQI	**Air quality conditions:**	**Symbol**
0 to 50	Good	Green
51 to 100	Moderate	Yellow
101 to 150	Unhealthy for Sensitive Groups	Orange
151 to 200	Unhealthy	Red
201 to 300	Very Unhealthy	Purple
301 to 500	Hazardous	Maroon

6.2 Deep learning

Deep learning is machine learning, but it functions more similarly to the human brain in a deeper and more advanced form. In other words, deep learning is part of a more prominent family of machine learning that focuses on methods based on artificial neural networks. This type of learning is an essential element in data science that receives raw inputs and extracts high-level features in several layers, including statistics and forecasting modeling. In-depth learning is very beneficial for data scientists to collect, analyze, and interpret large amounts of data. In general, it makes the process faster and easier. In other words, deep learning models process data more accurately and quicker due to the complexity and high ability to learn, especially in big data in research fields such as image processing, pattern recognition, and computer vision(Hochreiter & Urgen Schmidhuber, 1997). Deep learning is a powerful machine learning method that provides approximation, classification, and predictability (Gers et al., 2003)(Mohammadi et al., 2018). Figure 21 shows machine learning and deep learning (Bouloukian & Safi-Esfahani, 2020).

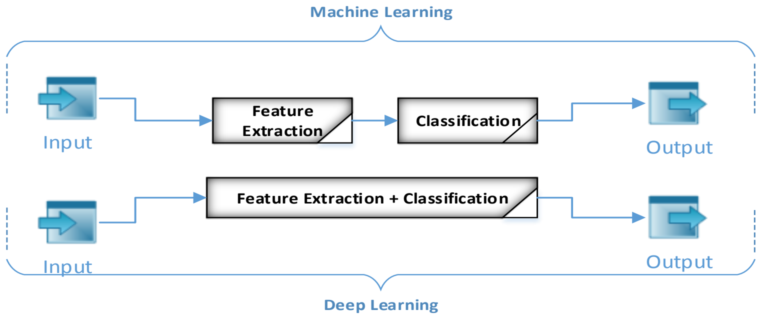


Figure 21: The difference between machine learning and deep learning(Boloukian & Safi-Esfahani, 2020)

6.3 Deep short-term neural network

The LSTM architecture [11] works better than conventional neural networks for long-term tasks [12], so it includes a deep and short-term neural network. Thus, the lower the short-term neural network output sequence, the higher the short-term neural network input sequence(Yao et al., 2014). The deep LSTM architecture used in(Graves et al., 2013)(Sharma et al.,

2018) can be described by ($\sigma(x)$, h_t^l , i_t^l , f_t^l , s_t^l , o_t^l), where l represents the layer index followed by Equation 7 to Equation 12. Equation 13 defines the Softmax activity function for a simple N-dimensional problem.

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{Equation 7}$$

$$h_t^l = o_t^l \tanh(S_t^l) \tag{Equation 8}$$

$$i_t^l = \sigma(W_i^l[X_t; h_{t-1}^l; h_t^{l-1}] + b_i^l \tag{Equation 9}$$

$$f_t^l = \sigma(W_f^l[X_t; h_{t-1}^l; h_t^{l-1}] + b_f^l \tag{Equation 10}$$

$$s_t^l = f_t^l s_{t-1}^l + i_t^l \tanh(W_s^l[X_t; h_{t-1}^l; h_t^{l-1}] + b_s^l) \tag{Equation 11}$$

$$o_t^l = \sigma(W_o^l[X_t; h_{t-1}^l; h_t^{l-1}] + b_o^l) \tag{Equation 12}$$

$$S_N = \{a \in R^N : a_i \in [0,1], \Sigma \sum^N a_i = 1\} \tag{Equation 13}$$

6.4 Generative Adversarial Networks | GAN

A generative adversarial network (GAN) is a class of machine learning. Generator modeling is an unsupervised learning activity in machine learning that automatically explores and learns the rules or patterns in input data. The model can generate or output new samples reliably retrieved from the original dataset. Generative adversarial networks are intelligent techniques to teach a productive model. They do this by framing the problem as a supervised learning problem with two sub-models: 1) Generator Model trained to produce new samples. 2) Discriminator Model, which tries to classify samples as actual (domain) or fake (produced). Both models train together over an adversarial zero-sum game. The parameters of the discriminate tool are set so that when the generator gives the actual data, it gets a high score, and when the data is fake, the generator score goes down. This process will continue until the model is not able to recognize them; In other words, the game continues until half of the time when the productive model produces believable examples(Sheng et al., 2020),

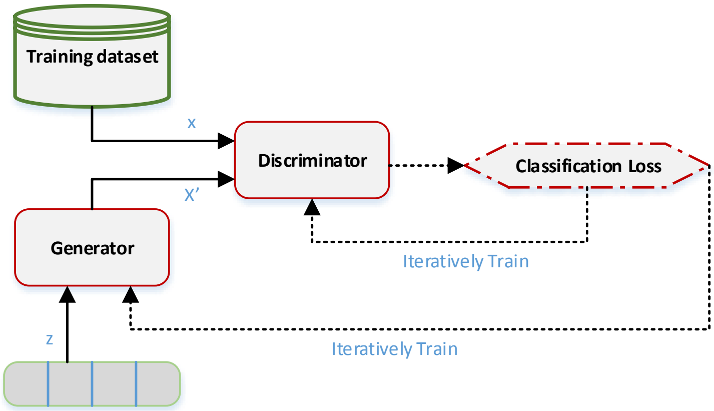
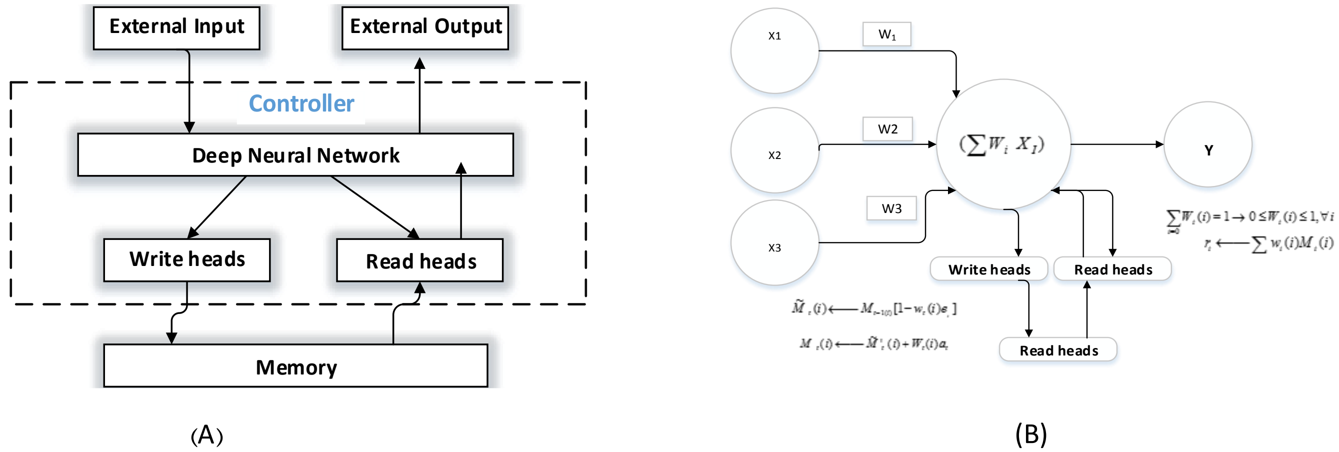


Figure 22: GAN network structure(Faradonbe & Safi-Esfahani, 2020)

6.5 Neural Turing Machine

It is a method derived from the Turing machine and neural networks. This model consists of Recurrent Neural networks (RNN) with an addressable external memory along with the ability of the recursive neural networks to perform algorithmic tasks such as sorting, copying, and N-gram. Generally speaking, a memory bank, a controller, and read-and-write heads are the main components of this method. The controller's job is to receive data from the outside and generate outputs during the update cycle. In addition, the Neural Turing Machine method guides the read-and-write heads directly into the external memory as a tape(Faradonbe & Safi-Esfahani, 2020)(Boloukian & Safi-Esfahani, 2020). Figure 23 shows the structure of the neural Turing machine in general in section(A) and its components in section(B)(Gulcehre et al., 2016).



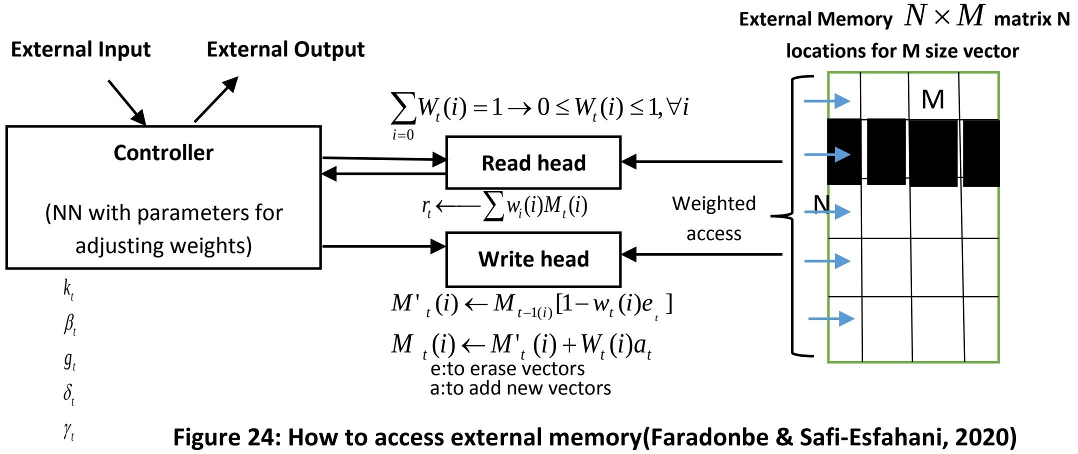
External Input
External Output
Controller
Deep Neural Network
Write heads
Read heads
Memory
(A)
x1
x2
x3
W1
W2
W3
$(\sum W_i X_I)$
Y
Write heads
Read heads
Read heads
$\sum W_t(i)=1 \to 0 \le W_t(i) \le 1, \forall i$
$r_t \longleftarrow \sum w_t(i) M_t(i)$
$\tilde{M}_t(i) \longleftarrow M_{t-1(i)}[1-w_t(i)e_t]$
$M_t(i) \longleftarrow \tilde{M}'_t(i)+W_t(i)a_t$
(B)

Figure 23: Structure of the neural Turing machine (A) in general (B) in part(Gulcehre et al., 2016)

6.5.1 The actions that the controller performs through the heads

The controller consists of five main functions: read, write, erase, move to the next memory cell, and move to the previous one. During each update cycle, the network controller receives inputs from the external environment and publishes the outputs in response. The network also starts reading and writing from a memory matrix using parallel read and write heads. The dotted line in Figure 3-A shows the division between the NTM circuit and the outside world. Each part of the structure is recognizable and distinct. This feature makes the network easier to train with gradient descent by defining "blurry or unclear" reading and writing operations. It also communicates more or less with all memory elements (instead of considering one element as a typical Turing machine or digital computer). A "focus" mechanism determines the degree of blur.

Each operation forces the read and writes to communicate with the small memory and ignore the rest. Because the interaction with memory is low and fragmented, NTM is based on data storage without interference. The output of the heads determines which memory location gets the most attention. These outputs define a normalized weighting on the rows of the memory matrix (referring to memory locations). At each assigned weight, each read/write head establishes the degree to which the head reads/writes in each area. Each head focuses precisely on a single memory location or multiple memory locations. Figure 24 displays the conceptual model of the controller's actions through the heads.

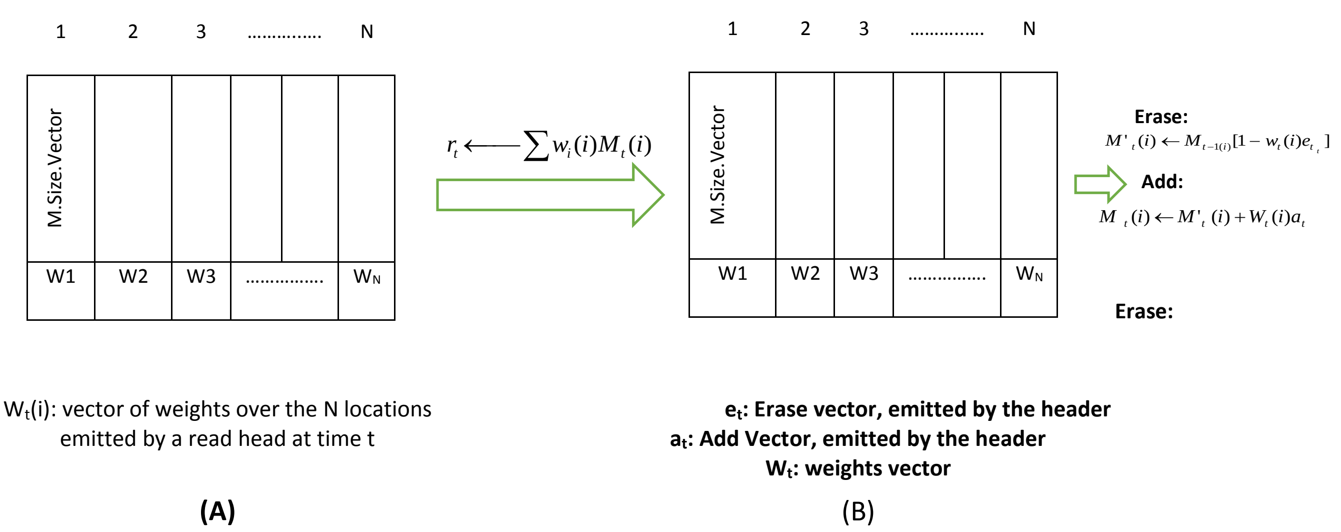


External Memory $N \times M$ matrix N
locations for M size vector
External Input
External Output
Controller
(NN with parameters for adjusting weights)
$\sum_{i=0} W_t(i)=1 \to 0 \le W_t(i) \le 1, \forall i$
Read head
$r_t \longleftarrow \sum w_i(i) M_t(i)$
Write head
Weighted access
M
N
$M'_t(i) \leftarrow M_{t-1(i)}[1-w_t(i)e_t]$
$M_t(i) \leftarrow M'_t(i)+W_t(i)a_t$
e:to erase vectors
a:to add new vectors
k_t
β_t
g_t
δ_t
γ_t

Figure 24: How to access external memory(Faradonbe & Safi-Esfahani, 2020)

6.5.2 Reading & Writing

The read and write operations are normalized weighting functions over the memory locations, similar to attention mechanisms. These weightings define a continuous distribution over the memory locations to make the operation differentiable. The reading operation is a simple linear combination of the memory locations:


1
2
3
....
N
M.Size.Vector
W1
W2
W3
....
WN
$r_t \longleftarrow \sum w_i(i) M_t(i)$
1
2
3
....
N
M.Size.Vector
W1
W2
W3
....
WN
Erase:
$M'_t(i) \leftarrow M_{t-1(i)}[1 - w_t(i)e_{t_t}]$
Add:
$M_t(i) \leftarrow M'_t(i) + W_t(i)a_t$
Erase:

$W_t(i)$: vector of weights over the N locations emitted by a read head at time t

e_t: Erase vector, emitted by the header
a_t: Add Vector, emitted by the header
W_t: weights vector

(A) (B)

Figure 25: Read and Write operation in NTM(Faradonbe & Safi-Esfahani, 2020)

Figure 25 shows the Read and Write operation in NTM (Faradonbe & Safi-Esfahani, 2020) in sections (A) and (B). Figure 25 – A shows the Nth of element $W_t(i)$ according to the limits of Equation 14:

$$\sum_{i=0} W_t(i) = 1 \rightarrow 0 \le W_t(i) \le 1, \forall i \qquad \text{Equation 14}$$

M_t is the contents of the M*N memory matrix at time t. Where N is the number of memory locations, and M is the size of the vector in each location. W is the vector of weights on the N locations propagated by a head reading at time t since all weights are normalized.

The read vector is a weighted convex combination of the memory location. The length M reads the return r_t vector by the head defined as a combination of row vectors $M_t(i)$ in memory, as shown in Equation 15, separable in terms of memory and weight.

$$r_t \longleftarrow \sum w_i(i) M_t(i) \qquad \text{Equation 15}$$

Figure 25 – B shows that the writing operation is a convex combination of erasing and writing to the memory locations. The write head outputs both erase (e) and add (a) vectors. Writing to the memory will then be made by erasing the locations defined by the write weighting vector, then adding to the locations specified by the same weighting vector. Again, notice that erasing and writing overall locations in different proportions make the whole operation differentiable, where parts of memory are erased according to the weighting vector, as shown in Equation 16:

$$M'_t(i) \leftarrow M_{t-1(i)}[1 - w_t(i)e_{t_t}] \qquad \text{Equation 16}$$

Multiplying it in the memory location works as a peer-to-peer. Therefore, the elements of the memory location are zeroed where the weight is, and the clearing element is 1. If the weight or clearance element is zero, the memory remains unchanged. When there are multiple writing heads, cleansing can be done in any order, where new information is added to locations defined by the weightings, as shown in Equation 17:

$$M_t(i) \leftarrow M'_t(i) + W_t(i)a_t \qquad \text{Equation 17}$$

Combining clearing and adding operations produce the final memory content at time t. Because deleting and adding are different, compound writing operations are also distinguishable. Note that both addition and subtraction vectors have M-independent components. Again, the function of the sequence of vectors added by multiple heads is trivial. Therefore, it allows precise control over the modified elements in each memory location.

6.5.3 Addressing mechanism

In the previous section, the equations of reading and writing were shown and examined. However, no explanation was given on how the weights are produced. These weights are created by combining two addressing mechanisms with complementary features. The first mechanism is content-based addressing, which looks at locations based on the similarity between current values and values published by the controller that addresses the content of Hopfield networks. The advantage of content-based addressing is that retrieval is easy. It only needs a controller to estimate a portion of the stored data to compare it to memory for accurate stored value. However, content-based addressing is not suitable for solving all problems. In some works, the content of a variable is as desired(Boloukian & Safi-Esfahani, 2020). However, the variable still needs a recognizable name or address. Computational problems are as follows: variables X and Y can take both values, but the $f(x, y) = x \times y$ trend must still be defined. A controller initializes the x and y variables, stores them in different addresses, retrieves them, and performs a multiplication algorithm. In this case, the variables are addressable by location, not content. This form of addressing is called location-based addressing. Content-based addressing is more commonly used than location-based addressing, as the content of a memory location can contain location information within it. However, it is necessary to provide location-based addressing as a primary operation for some generalized forms in experiments. As a result, both mechanisms are used together. The flow diagram of the addressing mechanism indicates the sequence of operations to construct a weight vector while reading and writing(Turing, 1950):

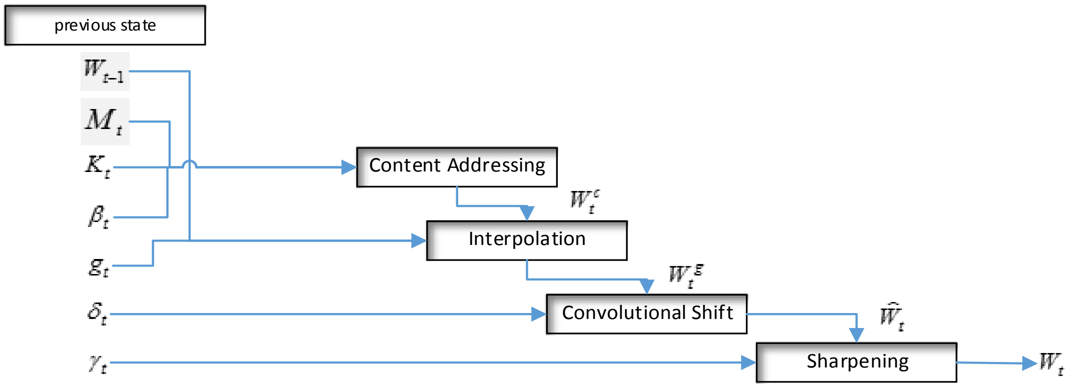


Figure 26: Flow diagram of the addressing mechanism (Turing, 1950)

6.5.4 Network controller

The NTM structure is described in the section. It has several free parameters: memory size, number of read/write heads, and allowable location changes. Perhaps the most important structure is the type of neural network used as the controller, especially when deciding whether to use a Recurrent or a Feedforward Neural Network (FNN). A recursive controller such as LSTM has internal memory to complete a larger matrix. Suppose a central processing unit controller on a digital computer (albeit with adaptive instructions instead of predefined ones) is compared to the RAM matrix. In that case, the hidden activities of the recursive controller are similar to the processors' registers. They allow the controller to combine information throughout the various time stages of the operation.

On the other hand, a feedforward controller mimics an RNN network by reading and writing to the specific memory location at each step. In addition, feedforward controllers often give more transparency to network operations. The read-and-write pattern on the memory matrix is interpreted easier than the internal state of an RNN. However, one of the limitations of a feedforward controller is that the number of read/write heads simultaneously causes a constraint (limiting operation) on the type of computation that NTM can perform. With a single read head, only a single conversion can be performed on a single memory vector at each time step. With two reading heads, it can do binary conversions, etc. Recurrent controllers can store reading vectors from previous efforts, so they do not suffer from this limitation.