الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
University of Badji Mokhtar Annaba
Faculty of Engineering Sciences
Department of Computer-Science

THEME :

LSTM Model for the Prediction of PM2.5 Concentration in city of Algiers

BY

Benteboula Chames Eddine

Thesis :
Submitted in partial fulfilment of the requirements for the degree of Master

Specialty :
Computer-Science System and Decision

Committee :

Dr. Sabri GHAZI	MCB	Supervisor	University Of Annaba
Dr. Mehdi Med Mendjel	MCB	President	University Of Annaba
Dr. Abdelouahid Alalga	MCB	Examiner	University Of Annaba

JULY - 2021

Dedications

Because he's the reason that we live and we shall die , today and always before anything we thank Almighty "ALLAH" so "الحمد لله" that i worked hard and made it this far .

To the woman that made me admire women , Thank you Mother for being the only spine that i lay on whenever I'm down , i would never imagine myself being here without your presence in my life , words can never describe it .

To my supervisor teacher that gave me a chance then guidance then gratitude and finally a work to remember.

And last but not least to my better half and bright side, i shall always be grateful for you bringing joy, love and motivation into my existence, from this day and till everything fades away .

Abstract

The work described in this thesis, aims to assess the performances of different LSTM (Long Short Term Memory) models to predict PM2.5 concentration. The models are trained using real data about Algiers city, the economic and politic capital of Algeria. In this study, we only used Measure about PM2.5 and we do not include climatic condition. A specific window from PM2.5 time series is used, the size of this window is varied in order to find the best one. A detailed presentation of related work is described, wherein; we put in light research conducted in order to build prediction models about air pollution generally and PM2.5 specifically. We tried several architecture of LSTM, by changing each time the parameters (batch size, number of stacked layer and the size of lagged values widow). For each architecture, we calculate the performances. We used RMSE (Root Mean Squared Error) in order to compare models' performances.

Résumé:

Le travail présenté dans ce document, concerne l'utilisation des modèles LSMT pour la prediction de la concetration du PM2.5. Plusieurs architectures ont été testées, afin de trouver la plus performante. Des mesures réelles ont été utilisées lors de l'apprentissage des modèles. Le parametrage des modèles a été fait en variant : le nombre de couches LSTM utilisées, Le types des couches LTSM utilisées, la taille du batch, ainsi que le nombre de valeurs passes. Le document contient un état de l'art qui décrit les recherches publiées recements qui traite ce sujet. Le performance des modèles est mesuré en utilisant distances : RMSE (la moyenne quatratique) pour comparer le performance des modèles.

ملخص

يهدف العمل الموصوف في هذه الأطروحة إلى تقييم أداء نماذج LSTM (الذاكرة طويلة المدى) المختلفة للتنبؤ بتركيز PM2.5. يتم تدريب النماذج باستخدام بيانات حقيقية عن مدينة الجزائر العاصمة الاقتصادية والسياسية للجزائر. في هذه الدراسة ، استخدمنا فقط قياس PM2.5 ولم نقم بتضمين الظروف المناخية. يتم استخدام نافذة محددة من السلسلة الزمنيةPM2.5 ، ويتنوع حجم هذه النافذة للعثور على أفضلها. يتم وصف عرض تفصيلي للأعمال ذات الصلة ، حيث ؛ أجرينا بحثًا خفيفًا من أجل بناء نماذج تنبؤ حول تلوث الهواء بشكل عام و PM2.5 على وجه التحديد. لقد جربنا العديد من بنية LSTM ، عن طريق تغيير المعلمات في كل مرة (حجم الدُفعة وعدد الطبقات المكدسة وحجم القيم المتأخرة). لكل هندسة نقوم بحساب الأداء. استخدمنا RMSE (خطأ مربع متوسط الجذر) لمقارنة أداء النماذج.

Contents :

CHAPTER III

CHAPTER IV

Figures :

Introduction

Over the last century, the global human society has augmented more than four times. Most of the recent increase is accredited to the public areas in the less refined parts of the world [1]. This has occurred in 80% of global cities and 98% of cities in low- and middle-income countries exceeding the recommendations for air quality [2]. Aside from economic losses, reduced clarity, and climate change, ambient air pollution costs millions of premature deaths annually, primarily due to anthropogenic slight particulate matter (PM2.5—particles with an aerodynamic width less than 2.5 μm) [3]. In the case of business-as-usual, the global climatic chemistry models suggest that the addition of outdoor air pollution to premature mortality could increase by 2050 [4].

Even though the concentrations of PM2.5 are 2–5 times higher in the developing countries, most of the air state studies and measurements are concentrated in the developed countries [5]. This is often due to the expenses required to launch and support a stable air quality monitoring station or network. High efficiency, standard air quality reference method material costs can range from $6000 to $36,000 per sensor [6], excluding the costs for support, calibration, and accessories, resulting in a price of an available air quality monitoring station well over $100,000. Meteorological equipment is also essential for evaluating air quality, as high UV transmission, high winds, rain, or extreme temperatures can cause serious health matters. Meteorological stations, depending on accuracy requirements, can cost from $1000 to over $7000; although, the accuracy differences are not too significant between the tiers (not including the lowest level equipment). Effective and nonhomogeneous urban systems contain different pollution sources, infrastructures, and varying terrains, requiring more than one station to comprehensively evaluate air pollution conditions, consequently excluding lower cities.

Recently, a different strategy aims at using machine learning to measure particulate pollution [7]. This study aims to evaluate the reliability for predicting air quality through a machine-learning strategy and from data sources with a separate scale of affordability. It centers on the case study of Quito, the capital city of Ecuador because it is a model example of complex terrain rapidly growing in mid-sized cities in the developing world with air pollution problems and economic limitations (e.g., poor quality fuel). In addition, Quito has several years of environmental data collection that can be used for data mining.

CHAPTER I

MACHINE LEARNING AND DEEP LEARNING

I MACHINE LEARNING AND DEEP LEARNING

1. Introduction

Data mining (DM) is the stage of the knowledge development process that aims to extract interesting and possibly useful information from data (Goodfellow et al., 2016; Mierswa, 2017). Although the term "data mining" is primarily oriented to large-scale data mining during this work, several techniques that function well for large-scale datasets can also be efficiently used to smaller datasets. Data mining can serve as a ground for Artificial Intelligence and Machine Learning. Various techniques in this area can be grouped in one of the fields .

2. How deep learning is different from machine learning and artificial intelligence

Artificial intelligence (AI)

It is a vast research field where machines show cognitive capabilities such as learning behaviors, proactive communication with the environment, inference and deduction, computer vision, language recognition, problem-solving, data representation, perception, and many others. More colloquially, AI means any activity where machines imitate intelligent behaviors typically shown by people. Artificial intelligence takes influence from elements of computer science, math, and statistics.

Machine learning (ML)

It is a subbranch of Al that focuses on teaching computers how to learn without needing to be programmed for specific tasks. The key concept behind ML is that it is possible to create algorithms that learn from and obtain predictions on data. There are three different broad classes of ML.

In supervised learning, the machine is performed with input data and desired output, and the goal is to learn from those training examples so that meaningful predictions can be made for new, unseen data. In unsupervised learning, the machine is acted with input data only, and 7the machine has to find some meaningful structure by itself with no external supervision. In reinforcement training, the machine acts as an agent interacting with the environment and learning the behaviors that generate rewards.

Deep learning (DL)

It is a particular subset of ML methodologies using artificial neural networks (ANN) somewhat caused by the structure of neurons located in the human brain. Informally, the term deep refers to many layers in the artificial neural network, but this meaning has changed over time. While four years ago, ten layers were already enough to consider a network as deep, today, it is more natural to consider a network as deep when it has hundreds of layers{x}.

3. Machine Learning

3.1 Introduction:

Machine learning is a set of mechanisms that allow us to "train" computers how to execute tasks by giving examples of how they should be done. For example, assume we wish to write a program to distinguish between actual email messages and unwanted spam. We could write a set of simple rules, such as flagging messages containing certain features (such as the word "viagra" or fake headers). However, writing rules to distinguish which text is valid accurately can be quite challenging to do well, resulting in either many missed spam messages or, worse, many lost emails. Worse, the spammers will actively adjust how they send spam to trick those strategies (e.g., writing "vi@gr@"). Writing practical rules — and keeping them up-to-date — quickly becomes an insurmountable task. Luckily, machine learning has granted a solution. New spam filters are "learned" from examples: we provide the learning algorithm with pattern emails which we have labeled as "ham" (valid email) or "spam" (unwanted email), and the algorithms learn to separate between them automatically.

3.2 Historical perspectives on machine learning

The investment in computational methods to learning dates back to artificial intelligence and cognitive science in the mid-1950s. The difference in both tasks and methods described research from the outset, including game playing, letter identification, abstract ideas, and verbal memory. Learning was viewed as a central feature of intelligent systems, and work on both learning and performance was concerned with developing general mechanisms for cognition, perception, and action.

In the 1960s, both AI researchers and psychologists recognized the importance of domain knowledge, which led to creating the first knowledge-intensive systems. However, learning researchers focused on general, domain-independent methods, with most work applied to perceptual domains. Eventually, pattern recognition and ar- artificial intelligence separated into two distinct fields. The gap widened further as many researchers in pattern memory began to emphasize algorithmic, numerical methods that contrasted sharply with the heuristic, symbolic methods compared with the AI paradigm.

During this period, most AI researchers avoided learning issues while they attempted to understand the role of knowledge in intelligent behavior. Research on knowledge representation, natural language, and expert systems dominated this era. However, some work on learning remained in the background, incorporating the designs and heuristic methods that had become central to artificial intelligence. Some work on concept learning and language property also occurred during this period.

In the 1970s, a new interest in machine learning emerged within AI and overgrew over a few years. Some research in this area was motivated by a frustration with the encyclopedic flavor and domain-specific emphasis of expert systems, as learning offered a return to general principles. Others were excited by the prospect of automating the acquisition of domain-specific knowledge bases, and still, others expected to model human learning. Research on data induction and language acquisition continued, but this was joined by work on machine discovery and learning in problem-solving. Several new methods were proposed, and a renewed interest in neural networks emerged, bringing back techniques abandoned by AI years earlier.

Machine learning continued to branch out during the 1980s, with work extending to planning, diagnosis, design, and control. Researchers became more severe about the real-world potential of learning algorithms, and several successful fielded applications showed that the technology could impact the industry. The field also placed itself on much firmer methodological grounds, with systematic experimentation on standard data sets and precise theoretical analysis becoming the norm rather than the exception. A variety of robust software packages also spread throughout the community, leading to careful comparative studies and a greater tendency to build on previous systems.

When the first workshop on machine learning took place at Carnegie Mellon University in 1980, only 30 participants were present. The field has developed rapidly since then, and it now boasts collected volumes on general and specialized topics, a refereed journal, and an annual conference that attracts some 300 participants. The number of papers on learning published in conference proceedings and journals has increased dramatically, and the number of PhD-level researchers in the area continues to grow. Machine learning has emerged as a central concern of the AI and cognitive science communities, as most researchers have come to acknowledge the central role of learning in intelligence.

3.3 Types of machine learning :

1) Supervised learning: Granted a training set of cases with suitable targets, and based on this training set, algorithms respond accurately to all feasible inputs. Learning from examples is another name of Supervised Learning. Classification and regression are the types of Directed Learning. Classification: It provides the prediction of Yes or No, example, "Is this tumor cancerous?", "Does this cookie match our class standards?" Regression: It answers "How much" and "How many."

2) Unsupervised learning: Correct answers or targets are not provided. The unsupervised learning technique tries to find out the similarities between the input data and these similarities, and the unsupervised learning technique classifies the data. This is also known as density estimation. Unsupervised learning contains clustering [8]. Clustering: it makes clusters based on similarity.

3) Semi-supervised learning: Semi-supervised training technique is a class of supervised training techniques. This learning also used unlabeled data for training purposes (generally, a minimum amount of labeled data with a massive amount of unlabeled data). Semi-supervised learning lies between unsupervised learning (unlabeled-data) and supervised training (labeled-data).

4) Reinforcement learning: This learning is supported by behaviorist psychology. The algorithm is informed when the answer is wrong but does not inform how to correct it. It has to examine and test various possibilities until it finds the correct solution. It is also known as learning with a judge. It does not suggest improvements. Reinforcement learning is different from supervised learning because accurate input and output sets are not offered, nor sub-optimal actions précised. Moreover, it focuses on online performance.

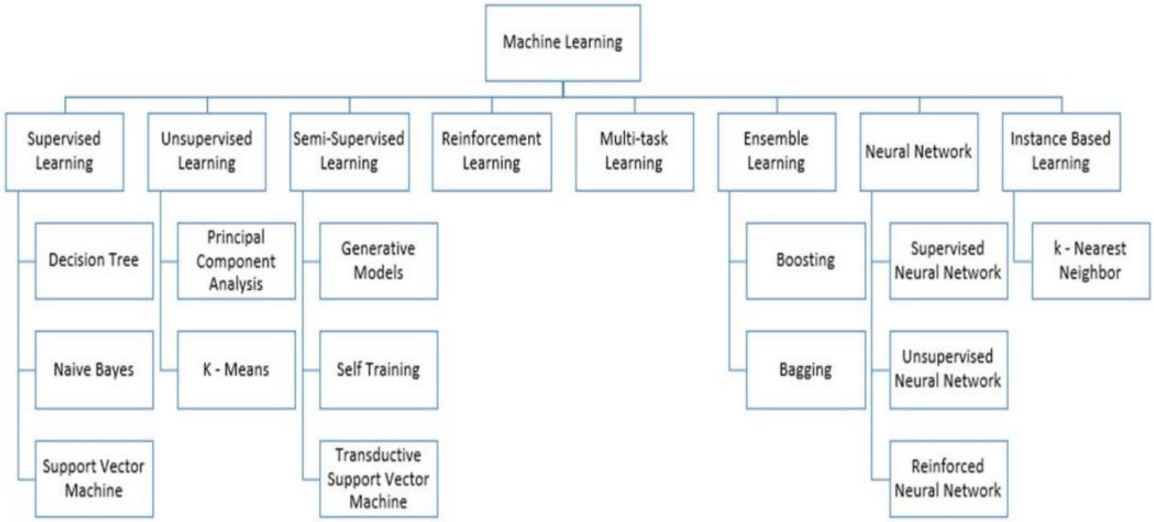


Figure 1:Types of Machine Learning

3.4 <u>**How machine learning works**</u>

There are four basic steps for building a machine learning application (or model). These are typically performed by data scientists working closely with the business professionals for whom the model is being developed.

Step 1: Select and prepare a training data set

Training data is a data set representative of the data the machine learning model will ingest to solve the problem it's designed to solve. In some cases, the training data is labeled data—'tagged' to call out features and classifications the model will need to identify. Other data is unlabeled, and the model will need to extract those features and assign classifications on its own. In either case, the training data needs to be properly prepared—randomized, de-duped, and checked for imbalances or biases that could impact the training. It should also be divided into two subsets: the training subset, which will be used to train the application, and the evaluation subset, used to test and refine it.

Step 2: Choose an algorithm to run on the training data set

Again, an algorithm is a set of statistical processing steps. The type of algorithm depends on the type (labeled or unlabeled) and amount of data in the training data set and on the type of problem to be solved.

Step 3: Training the algorithm to create the model

Training the algorithm is an iterative process–it involves running variables through the algorithm, comparing the output with the results it should have produced, adjusting weights and biases within the algorithm that might yield a more accurate result, and running the variables again until the algorithm returns the correct result most of the time. The resulting trained, accurate algorithm is the machine learning model—an important distinction to note, because 'algorithm' and 'model' are incorrectly used interchangeably, even by machine learning mavens.

Step 4: Using and improving the model

The final step is to use the model with new data and, in the best case, for it to improve in accuracy and effectiveness over time. Where the new data comes from will depend on the problem being solved. For example, a machine learning model designed to identify spam will ingest email messages, whereas a machine learning model that drives a robot vacuum cleaner will ingest data resulting from real-world interaction with moved furniture or new objects in the room.

3.5 Types of machine learning algorithms

- **Regression algorithms:**

  Linear and logistic regression are examples of regression algorithms used to understand relationships in data. Linear regression is used to predict the value of a dependent variable based on the value of an independent variable. Logistic regression can be used when the dependent variable is binary in nature: A or B. For example, a linear regression algorithm could be trained to predict a salesperson's annual sales (the dependent variable) based on its relationship to the salesperson's education or years of experience (the independent variables.) Another type of regression algorithm called a support vector machine is useful when dependent variables are more difficult to classify.

- **Decision trees:**

  Use classified data to make recommendations based on a set of decision rules. For example, a decision tree that recommends betting on a particular horse to win, place, or show could use data about the horse (e.g., age, winning percentage, pedigree) and apply rules to those factors to recommend an action or decision.

- **Instance-based algorithms:**

  A good example of an instance-based algorithm is K-Nearest Neighbor or k-nn. It uses classification to estimate how likely a data point is to be a member of one group or another based on its proximity to other data points.

- **Clustering algorithms:**

  Think of clusters as groups. Clustering focuses on identifying groups of similar records and labeling the records according to the group to which they belong. This is done without prior knowledge about the groups and their characteristics. Types of clustering algorithms include the K-means, TwoStep, and Kohonen clustering.

- **Association algorithms:**

  Association algorithms find patterns and relationships in data and identify frequent 'if-then' relationships called *association rules*. These are similar to the rules used in data mining.

- **Neural networks:**

  A neural network is an algorithm that defines a layered network of calculations featuring an input layer, where data is ingested; at least one hidden layer, where calculations are performed make different conclusions about input; and an output layer. where each conclusion is assigned a probability. A deep neural network defines a network with multiple hidden layers, each of which successively refines the results of the previous layer.

4. Deep learning

4.1 **Introduction**

Deep learning is the subfield of artificial intelligence that focuses on building large neural network models capable of making accurate data-driven decisions. Deep learning particularly suited to contexts where the data is complex and large datasets are available. Today most online companies and high-end consumer technologies use deep learning. Among other things, Facebook uses deep learning to analyze text in online conversations. Google, Baidu, and Microsoft all use deep learning for im- age search and machine translation. All modern smartphones have deep learning systems running on them; for example, deep learning is now the standard technology for speech recognition and de- face section on digital cameras. In the healthcare sector, deep learning used to process medical images (X-rays, CT, and MRI scans) and diagnose health conditions. Deep learning is also at the core of self-driving cars, where it is used for localization and mapping, motion planning and steering, environment perception, and tracking driver states.{x}.

4.2 **What is Neural Network?**

Neural Networks are algorithms that closely match the human brain and are designed to recognize patterns. They interpret sensory data in a machine perception, labeling, or raw clustering input. They can recognize numerical patterns in vectors, into which all real-world data (images, sound, text, or time series) must be translated. Artificial neural networks are formed of many highly interconnected processing elements (neurons) working together to resolve a problem.
An ANN usually involves a large amount of processors operating in parallel and arranged in tiers. The first tier receives the basic input information — similar to optic nerves in human visual

processing. Each successive tier receives the product from the tier leading it rather than from the raw input — in the same way, neurons extra from the optic nerve receive signals from those closest to it. The last tier provides the output of the system [9].

4.3 **Recurrent Neural Network (RNN)**

4.3.1 What is Recurrent Neural Network (RNN) ?

The recurrent neural network was developed in the 1980s [10–11]. Its construction consists of an input layer, one or more hidden layers, and an output layer. RNNs have chain-like structures of repeating modules with the idea behind using these modules as a memory to store important information from earlier processing steps. Unlike feedforward neural networks, RNNs add a feedback loop that allows the neural network to accept a series of inputs. This means the output from step t − 1 is fed back into the network to determine the outcome of step t and for each subsequent step. Consequently, RNNs have been successful in learning sequences.

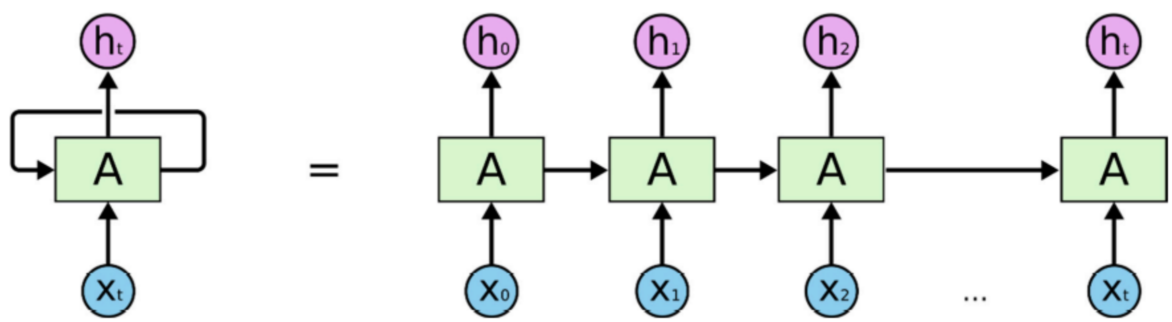


Figure 2:Sequential processing in RNN.

Figure 2 shows a simple RNN with one input unit, one output unit, and one recurrent hidden unit expanded into a complete network, where Xt is the input at time step t and ht is the output at time step t. RNN uses a backpropagation algorithm during the training process and a prevalent algorithm used in calculating gradients and adjusting weight matrices in ANN. However, it will adjust and update the powers following the modification of the feedback process. Therefore, it is commonly related to as the backpropagation through time (BPTT). The BPTT process uses a working-backward path, layer by layer, from the network's final output, tweaking the powers of each unit according to the unit's estimated portion of the total amount error. The information loops repeat, resulting in considerable updates to neural network model weights and lead to an unstable network due to the growth of error gradients during the updating process. Therefore, BPTT is not enough efficient to learn a pattern from long-term dependency because of the gradient vanishing and the exploding gradient problems [12]. This would be one of the important reasons leading to difficulties in the training of recurrent neural networks [13–14].

4.3.2 Advantages of Recurrent Neural Network

1. RNN can model a sequence of data so that each sample can be assumed to be dependent on previous ones

2. Recurrent neural networks are even used with convolutional layers to extend the effective pixel neighborhood.

4.3.3 Disadvantages of Recurrent Neural Network

1. Gradient vanishing and exploding problems.

2. Training an RNN is a challenging task.

3. It cannot process very long sequences if using *tanh* or *relu* as an activation function.

4.4 **Long Short-Term Memory (LSTM) Neural Network**

4.4.1 What is Long Short Term Memory (LSTM)?

Long Short-Term Memory, an evolving of RNN, was founded by Hochreiter and Schmidhuber [14] to address the drawbacks above of the RNN by adding further cooperations per module (or cell). LSTMs are a unique kind of RNN, capable of learning long-term dependencies and remembering information for prolonged periods as a default. According to Olah [15], the LSTM model is designed in a chain structure. However, the repeating module has a composite structure. Instead of a single neural network like a regular RNN, it has four interacting layers with a single method of communication.

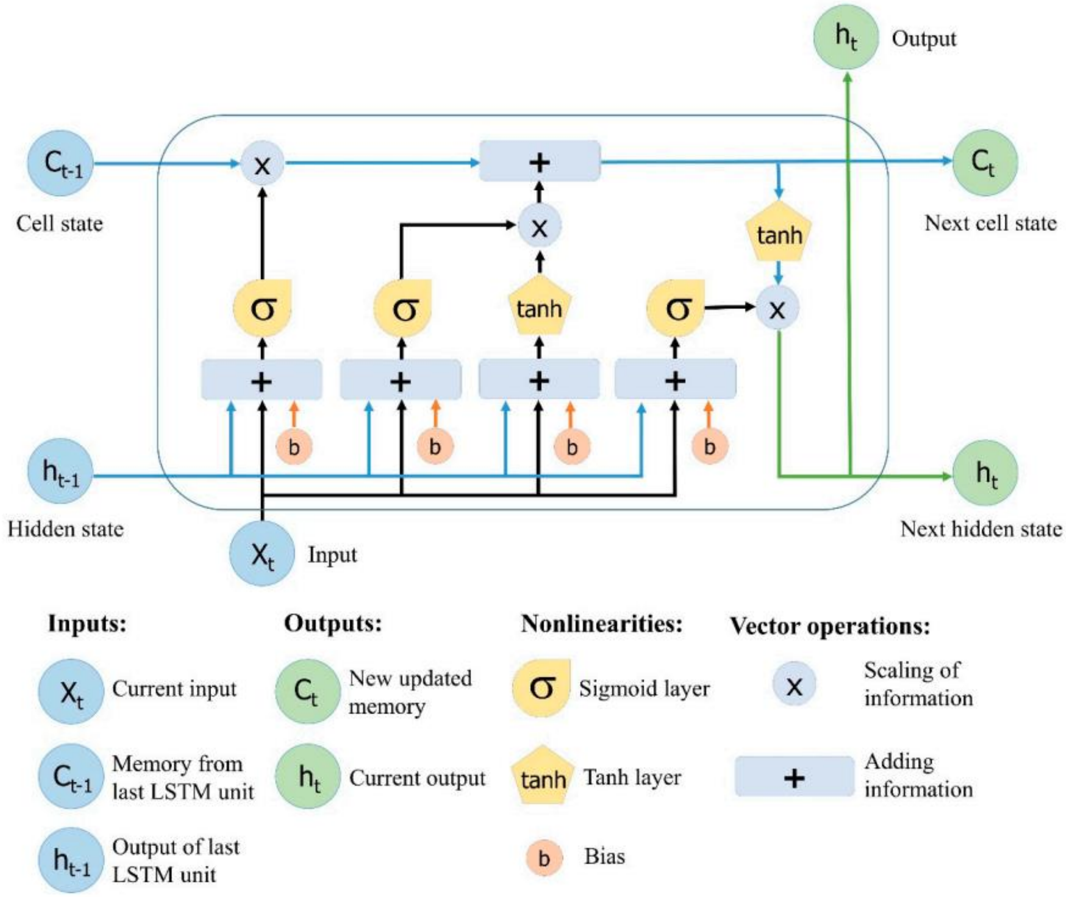
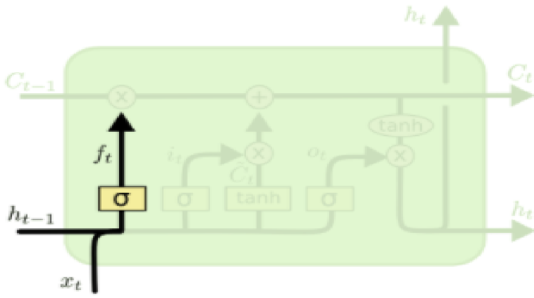


Figure 3:The structure of the long short-term memory (LSTM) neural network.

A typical LSTM network is composed of memory blocks called cells. Two cases are being transferred to the next cell, the cell state and the hidden state. The cell state is the primary data flow chain, which allows the data to flow forward virtually unchanged. However, some linear changes may occur. The data can be added to or removed from the cell state via sigmoid gates. A gate is related to a layer or a series of matrix operations containing many individual weights. LSTMs are intended to avoid the long-term dependency problem because it uses gates to control the memorizing process.

4.4.2 LSTM WalkThrough

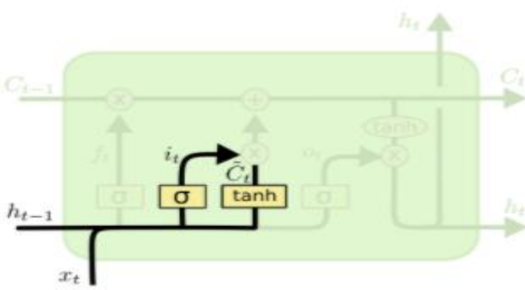
The first step in constructing an LSTM network is to identify information that is not required and will be omitted from the cell in that step. This process of identifying and excluding data is decided by the sigmoid function, which takes the output of the last LSTM unit (h_{t-1}) at time $t-1$ and the current input (X_t) at time t. Additionally, the sigmoid function determines which part from the old output should be eliminated. This gate is called the forget gate (or f_t); where f_t is a vector with values ranging from 0 to 1, corresponding to each number in the cell state, C_{t-1}.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Herein, σ is the sigmoid function, and Wf and bf are the weight matrices and bias, respectively, of the forget gate.

The following step is deciding and storing information from the new input (Xt) in the cell state as well as to update the cell state. This step contains two parts, the sigmoid layer and second the tanh layer. First, the sigmoid layer decides whether the new information should be updated or ignored (0 or 1), and second, the tanh function gives weight to the values which passed by, deciding their level of importance (−1 to 1). The two values are multiplied to update the new cell state. This new memory is then added to old memory Ct−1 resulting in Ct .

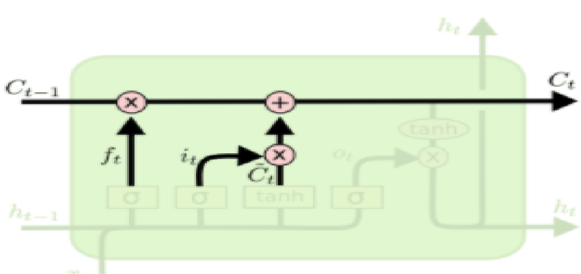


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

It's now time to update the old cell state, Ct−1Ct−1, into the new cell state CtCt. The previous steps already decided what to do, we just need to actually do it.

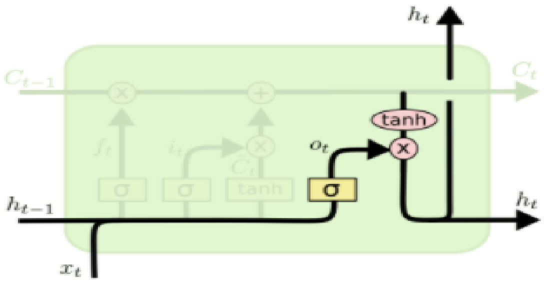
We multiply the old state by ftft, forgetting the things we decided to forget earlier. Then we add it∗C~t it∗C~t. This is the new candidate values, scaled by how much we decided to update each state value.

In the case of the language model, this is where we'd actually drop the information about the old subject's gender and add the new information, as we decided in the previous steps



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

In the final step, the output values (ht) is based on the output cell state (Ot) but is a filtered version. First, a sigmoid layer decides which parts of the cell state make it to the output. Next, the output of the sigmoid gate (Ot) is multiplied by the new values created by the tanh layer from the cell state (Ct), with a value ranging between −1 and 1.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Here, Wo and bo are the weight matrices and bias, respectively, of the output gate.

4.4.3 <u>Advantages of LSTM :</u>

- The constant error backpropagation within memory cells results in LSTM's ability to bridge very long time lags in case of problems related to those discussed above.
- For long-time lag problems such as those discussed in this article, LSTM can handle noise, distributed representations, and continuous values. In difference to finite-state automata or hidden Markov models, LSTM does not require an a priori choice of a limited number of states. In principle, it can deal with unlimited state numbers.
- For problems discussed in this article, LSTM generalizes well, even if the positions of widely separated, relevant inputs in the input sequence do not matter. Unlike previous approaches, ours quickly learns to distinguish between two or more widely separated occurrences of a particular element in an input sequence, without depending on ap[1]propriate short-time-lag training exemplars.
- There appears to be no need for fine parameter tuning. LSTM works fine over a wide range of parameters such as learning rate, input gate bias, and output gate bias. For example, the learn[1]ing rates used in our experiments may seem enormous to some readers. However, a large learning rate pushes the output gates toward zero, thus automatically countermanding its adverse effects.
- The LSTM algorithm's update complexity per weight and time step is essentially that of BPTT, namely, O (1). This is great in comparison to other methods such as RTRL. Unlike full BPTT, however, LSTM is local in both space and time.

4.5 **Applications of LSTM-RNN**

1) Early learning tasks

In early experiments, LSTM proved applicable to varied learning tasks, pre_viously considered impossible to find out. This included recalling high precision real numbers over extended noisy sequences [16], learning context-free lan_guages [17], and various tasks that need precise timing and counting [18]. In [19], LSTM was successfully introduced to meta-learning with program search tasks to approximate a learning algorithm for quadratic functions. The successful application of reinforcement learning to resolve non-Markovian learning tasks with long-term dependencies was shown by [20].

2) Cognitive learning tasks

LSTM-RNNs proved great strengths in solving an oversized kind of cognitive learn_ing tasks. Speech and handwriting recognition, and more recently, artificial intelligence, are the foremost predominant in literature. Other cognitive learn_ing tasks include emotion recognition from speech [21], text generation [22], handwriting generation [23], constituency parsing [24], and conversational mod_elling [25].

3) Speech recognition

A first indication of the capabilities of neural networks in tasks associated with nat_ural language was given by [26] with a neural language modeling task. In 2003 good results applying standard LSTM-RNN networks with a combination of LSTM and sigmoidal units to speech recognition tasks were obtained by .Better results adore Hidden-Markov-Model (HMM)-based systems [27] were achieved using bidirectional training with BLSTM [28]. A variant named BLSTM-CTC [29] finally outperformed HMMs, with recent improve_ments documented in [30]. A deep variant of stacked BLSTM-CTC was utilized in 2013 by [31] and later extended with a modified CTC objective func_tion by [30], both achieving outstanding results. The performance of various LSTM-RNN architectures on large vocabulary speech recognition tasks was in_vestigated by [32], with best results using an LSTM/HMM hybrid architecture. Comparable results were achieved by.

More recently LSTM was improving results using the sequence-to-sequence framework ([33]) and attention-based learning .In 2015 [34] introduced a specialized architecture for speech recognition with two functions, the primary called 'listener' and therefore the latter called 'attend and spell.' The 'listener' function uses BLSTM with a pyramid structure (pBLSTM), just like clockwork RNNs introduced by [35]. the opposite function, 'attend and spell,' uses an attention_based LSTM transducer developed by and .Both functions are trained with methods introduced within the sequence-to-sequence framework and attention-based learning [36].

4) Handwriting recognition

In 2007 introduced BLSTM-CTC and applied it to online handwriting recog_nition, leading to outperforming Hidden-Markov-based recognition sys_tems presented by [37]. combined BLSTM-CTC with a probabilistic lan_guage model and by this developed a system capable of directly transcribing raw online handwriting data. in a very real-world use case, this technique showed a high automation rate with a slip-up rate akin to an individual is on this type of task [38]. In another approach combined BLSTM-CTC with multi-dimensional LSTM and applied it to an offline handwriting recognition task, yet outper_forming classifiers supported Hidden-Markov models. In 2013 applied the very successful regularisation method dropout as proposed by [39]).

5) Machine translation

In 2014 , the authors applied the RNN encoder-decoder neural network ar_chitecture to computational linguistics and improved the performance of a statistical, computational linguistics system. The RNN Encoder-Decoder architecture is predicated on an approach communicated by [40]. a similar deep LSTM architecture, spoken as sequence-to-sequence learning, was investigated by , confirming these results. addressed the rare word problem using sequence-to-sequence, which improves the power to translate words, not vocabulary. The archi_tecture was further improved by [41] addressing issues associated with the interpretation of long sentences by implementing an attention mechanism into the decoder.

6) Image processing

In 2012, BSLTM was applied to keyword spotting and mode detection, distin_guishing differing types of content in handwritten documents, like text, formulas, diagrams, and figures, outperforming HMMs and SVMs [42]. At approximately the identical period investigated the classification of high-resolution images from the ImageNet database with many better results than previous approaches. In 2015 the newer LSTM variant using the Sequence-to-Sequence framework was successfully trained by [43] to get natural sentences in plain English describing images. Also, in 2015, the authors combined LSTMs with a deep hierarchical visual feature extractor and applied the model to image interpretation and classification tasks, like activity recognition and image/video description.

5. Conclusion

In this chapter we had a overview on the what is machine learning and deep learning and what's the difference between them , and then we explained how machine learning works and the diffrent types of algorithms used on this domain .In the second part we saw two types of deep learning (RNN,LSTM) which are basically the same but LSTMs are often referred to as fancy RNNs because they have both cell states and a hidden states ,and that includes also seeing some of the advantages and disadvantages of both algorithms .

CHAPTER II

STATE OF THE ART

II STATE OF THE ART

1. Introduction

One of our era's greatest scourges is air pollution, on account not only of its impact on climate change but also its impact on public and individual health due to increased morbidity and mortality. Many pollutants are major factors in disease in humans. Among them, Particulate Matter (PM), particles of variable but very small diameter, penetrate the respiratory system via inhalation, causing respiratory and cardiovascular diseases, reproductive and central nervous system dysfunctions, and cancer. Even though ozone in the stratosphere plays a protective role against ultraviolet irradiation, it is harmful when in high concentration at ground level, also affecting the respiratory and cardiovascular system. Furthermore, nitrogen oxide, sulfur dioxide, Volatile Organic Compounds (VOCs), dioxins, and polycyclic aromatic hydrocarbons (PAHs) are all considered air pollutants that are harmful to humans. Carbon monoxide can even provoke direct poisoning when breathed in at high levels. Heavy metals such as lead, when absorbed into the human body, can lead to direct poisoning or chronic intoxication, depending on exposure. Diseases occurring from the aforementioned substances include principally respiratory problems such as Chronic Obstructive Pulmonary Disease (COPD), asthma, bronchiolitis, and also lung cancer, cardiovascular events, central nervous system dysfunctions, and cutaneous diseases. Last but not least, climate change resulting from environmental pollution affects the geographical distribution of many infectious diseases, as do natural disasters. The only way to tackle this problem is through public awareness coupled with a multidisciplinary approach by scientific experts; national and international organizations must address the emergence of this threat and propose sustainable solutions[44].

2. Climate and Pollution

Air pollution and global climate change are closely related. Climate is that the other side of the identical coin reduces the standard of our Earth. Pollutants like black carbon, methane, tropospheric ozone, and aerosols affect incoming sunlight. As a result, the Earth's temperature increases, leading to the melting of ice, icebergs, and glaciers.

During this vein, climatic changes will affect the rate and prevalence of both residual and imported infections in Europe. Climate and weather modify the duration, timing, and intensity of outbreaks strongly and alter the map of infectious diseases worldwide (45). Mosquito-transmitted parasitic or viral diseases are highly climate-sensitive, as warming firstly shortens the pathogen period and also shifts the geographic map of the vector. Similarly, water-warming following climate changes result in a high incidence of waterborne infections. Recently, in Europe, eradicated diseases seem to be emerging because of the migration of the population, for instance, cholera, poliomyelitis, tick-borne encephalitis, and malaria (46).

The spread of epidemics is related to natural climate disasters and storms, which seem to occur more frequently (47). Malnutrition and disequilibration of the system are related to the emerging infections affecting public health (48).

The Chikungunya virus "took the airplane" from the Indian Ocean to Europe, as outbreaks of the disease were registered in Italy (49) similarly as autochthonous cases in France (50).
An increment in cryptosporidiosis in the United Kingdom and the Czech Republic appears to have occurred following flooding (51). As said previously, aerosols composites are tiny in size and considerably affect the climate. They can consume sunlight (the albedo phenomenon) by dispersing a quarter of the sun's rays back to space and have cooled the global temperature over the last 30 years (52).

3. Air Pollutants

The World Health Organization (WHO) reports six major air pollutants: particle pollution, ground-level ozone, carbon monoxide, sulfur oxides, nitrogen oxides, and lead. Air pollution can have catastrophic effects on all environmental components, including groundwater, soil, and air. Also, it poses a serious threat to living organisms. Therefore, our primary interest is in these pollutants, as they are associated with more extensive and severe human health and environmental pollution problems. Acid rain, global warming, the greenhouse effect, and climate change have a significant environmental impact on air pollution (53).

4. Particulate Matter (PM)

4.1 **What is particulate matter?**

PM could be a widespread air pollutant, consisting of a combination of solid and liquid particles suspended within the air. Commonly used indicators describing PM relevant to health discuss with the mass concentration of particles with a diameter of but 10 µm (PM10) and particles with a diameter of but 2.5 µm (PM2.5). PM2.5, often called fine PM, also comprises ultrafine particles having a diameter of but 0.1 µm. In most locations in Europe, PM2.5 constitutes 50–70% of PM10. PM between 0.1 µm and 1 µm in diameter can remain within the atmosphere for days or weeks and thus be subjected to long-range transboundary transport within the air. PM may be a mixture of physical and chemical characteristics varying by location. Common chemical constituents of PM include sulfates, nitrates, ammonium, other inorganic ions like ions of sodium, potassium, calcium, magnesium, and chloride, organic and elemental carbon, crustal material, particle-bound water, metals (including cadmium, copper, nickel, vanadium, and zinc) and polycyclic aromatic hydrocarbons (PAH). Additionally, biological components like allergens and microbial compounds are found in PM.

4.2 **PM And Health**

Studies have revealed a relationship between particulate matter (PM) and adverse health effects, centering on either short-term (acute) or long-term (chronic) PM exposure.

Particulate matter (PM) is regularly made in the climate due to chemical effects between the different pollutants. The entrance of particles is closely dependent on their size (53). Particulate Matter (PM) was set as a term for particles by the United States Environmental Protection Agency (54). Particulate matter (PM) pollution includes particles with diameters of 10 micrometers (µm) or smaller, called PM10, and fabulous particles with diameters that are generally 2.5 micrometers (µm) and smaller.

Particulate matter contains small liquid or solid droplets that can be sniffed and cause serious health effects (55). Particles <10µm in diameter (PM10) after inhalation can invade the lungs and even reach the bloodstream. Fine particles, PM2.5, pose a greater risk to health (56) .

Particle size	Penetration degree in human respiratory system
>11 µm	Passage into nostrils and upper respiratory tract
7–11 µm	Passage into nasal cavity
4.7–7 µm	Passage into larynx
3.3–4.7 µm	Passage into trachea-bronchial area
2.1–3.3 µm	Secondary bronchial area passage
1.1–2.1 µm	Terminal bronchial area passage
0.65–1.1 µm	Bronchioles penetrability
0.43–0.65 µm	Alveolar penetrability

Figure 4:Penetrability according to particle size.

Multiple epidemiological studies are performed on the health effects of PM. A positive relation was shown between short-term and long-term exposures of PM2.5 and acute nasopharyngitis (56). additionally, long-term exposure to PM for years was found to be associated with cardiovascular diseases and mortality.

Those studies rely on PM2.5 monitors and are restricted in terms of the study area or city area because of a scarcity of spatially resolved daily PM2.5 concentration data and, during this way, are not representative of the whole population. Following a recent epidemiological research by the Department of Environmental Health at Harvard School of Public Health (Boston, MA) (57), it had been reported that, as PM2.5 concentrations change spatially, an exposure error (Berkson error) seems to be created, and also the relative magnitudes of the short- and long-term effects are not yet fully elucidated. The team developed a PM2.5 exposure model that supported remote sensing data for assessing short- and long-term human exposures (57). This model permits spatial resolution in short-term effects plus the assessment of long-term effects within the whole population.

Moreover, respiratory diseases and affection of the system are registered as long-term chronic effects (58). it is worth noting that individuals with asthma, pneumonia, diabetes and respiratory and cardiovascular diseases are especially susceptible and liable to the consequences of PM. PM2.5, followed by PM10, is strongly related to diverse systema respiratorium diseases (59), as their size permits them to pierce interior spaces (60). The particles produce toxic effects in keeping with their chemical and physical properties. The components of PM10 and PM2.5 are often organic (polycyclic aromatic hydrocarbons, dioxins, benzene, 1-3 butadiene) or inorganic (carbon, chlorides, nitrates, sulfates, metals) in nature (55).

Material (PM) is split into four main categories in line with type and size (61) (Table 2). Gas contaminants include PM in aerial masses. Particulate contaminants include contaminants like smog, soot, tobacco smoke, oil smoke, fly ash, and cement dust.

Type		PM diameter [μm]
Particulate contaminants	Smog	0.01–1
	Soot	0.01–0.8
	Tobacco smoke	0.01–1
	Fly ash	1–100
	Cement Dust	8–100
Biological Contaminants	Bacteria and bacterial spores	0.7–10
	Viruses	0.01–1
	Fungi and molds	2–12
	Allergens (dogs, cats, pollen, household dust)	0.1–100
Types of Dust	Atmospheric dust	0.01–1
	Heavy dust	100–1000
	Settling dust	1–100
Gases	Different gaseous contaminants	0.0001–0.01

Figure 5:Types and sizes of particulate matter (PM).

Biological Contaminants are microorganisms (bacteria, viruses, fungi, mold, and bacterial spores), cat allergens, home dust and allergens. Types of Dust include dissolved atmospheric dust, settling dust, and heavy dust. Lastly, another fact is that the half-lives of PM10 and PM2.5 particles in the atmosphere is increased due to their tiny dimensions; this permits their stable delay in the atmosphere and even their transfer and spread to distant targets where people and the environment may be exposed to the same magnitude of pollution. They can change the nutrient balance in watery ecosystems, damage forests, and crops, and acidify water bodies. As stated, PM2.5, due to its tiny size, is causing more severe health effects. These aforementioned fine particles are the primary cause of the "haze" formation in different metropolitan areas (61).

4.3 <u>**Table of pm and health outcomes:**</u>

HEALTH OUTCOMES	SHORT-TERM STUDIES			LONG-TERM STUDIES		
	PM_{10}	$PM_{2.5}$	UFP	PM_{10}	$PM_{2.5}$	UFP
Mortality						
All causes	XXX	XXX	X	XX	XX	X
Cardiovascular	XXX	XXX	X	XX	XX	X
Pulmonary	XXX	XXX	X	XX	XX	X
Pulmonary effects						
Lung function, eg, PEF	XXX	XXX	XX	XXX	XXX	
Lung function growth				XXX	XXX	
Asthma and COPD exacerbation						
Acute respiratory symptoms		XX	X	XXX	XXX	
Medication use			X			
Hospital admission	XX	XXX	X			
Lung cancer						
Cohort				XX	XX	X
Hospital admission				XX	XX	X
Cardiovascular effects						
Hospital admission	XXX	XXX		X	X	
ECG-related endpoints						
Autonomic nervous system	XXX	XXX	XX			
Myocardial substrate and vulnerability		XX	X			
Vascular function						
Blood pressure	XX	XXX	X			
Endothelial function	X	XX	X			
Blood markers						
Pro-inflammatory mediators	XX	XX	XX			
Coagulation blood markers	XX	XX	XX			
Diabetes	X	XX	X			
Endothelial function	X	X	XX			
Reproduction						
Premature birth	X	X				
Birth weight	XX	X				
IUR/SGA	X	X				
Fetal growth						
Birth defects	X					
Infant mortality	XX	X				
Sperm quality	X	X				
Neurotoxic effects						
Central nervous system		X	XX			

Note : (X) = few studies. (XX) = many studies. (XXX) = large number of studies.
Abbreviation : (UFP) = ultrafine particle ,(PEF) = peak expiratory flow ,(COPD) = chronic obstructive pulmonary disease ,(IUG) = intrauterine growth restricion ,(SAG) = small for gestational age .

4.4 **Environmental Effects**

The primary significant environmental effects are as follows:

Acid rain is wet (rain, fog, snow) or dry (particulates and gas) precipitation containing toxic amounts of nitric and sulfuric acids. They will acidify the water and soil conditions, damage trees and farms, and even damage buildings and outdoor sculptures, constructions, and statues.

Haze is produced when fine particles are dispersed within the air and reduce the transparency of the atmosphere. It's caused by gas emissions within the air from industrial plants, power plants, cars, and trucks.

Ozone, as presented previously, occurs both at ground level and within the upper level (stratosphere) of the Earth's climate. Stratospheric ozone is protecting us from the Sun's harmful ultraviolet (UV) rays. In contrast, ground-level ozone is toxic to human health and may be a pollutant. Sadly, stratospheric ozone is gradually damaged by ozone-depleting substances (i.e., chemicals, pesticides, and aerosols). If this protecting stratospheric layer is thinned, then UV radiation can reach our Earth, with harmful effects on human life (skin cancer) (74) and crops (75). In flowers, ozone penetrates through the stomata, inducing them to shut, which blocks CO2 transfer and induces a discount in photosynthesis (76).

Global temperature change is a vital issue that concerns humanity. As is, though, the "greenhouse effect" keeps the Earth's heat stable. Unhappily, anthropogenic activities have destroyed this protecting heat effect by producing large amounts of greenhouse gases. Heating is mounting, with harmful effects on human health, animals, forests, wildlife, agriculture, and the water environment. A report states that warming is adding to the health risks of poor people (77).
People living in poorly constructed structures in warm-climate nations are at high risk for heat-related health problems as heats mount (77).

Wildlife is overwhelmed by toxic pollutants coming from the air, soil, or the water ecosystem. In this way, animals can improve health problems when exposed to high levels of pollutants. Reproductive failure and birth effects are reported. Eutrophication occurs when elevated concentrations of nutrients (especially nitrogen) stimulate the blooming of aquatic algae, which may cause a disequilibration within the diversity of fish and their deaths.

Without a doubt, there's a critical concentration of pollution that an ecosystem can tolerate without being destroyed, which is related to the ecosystem's capacity to neutralize acidity. The Canada air pollution Program established this load at 20 kg/ha/yr (78).

Hence, pollution has harmful effects on both soil and water (79).Regarding PM as an air pollutant, its impact on crop yield and food production has been reported. Its impact on watery bodies is related to the survival of living organisms and their productivity potential (79).

An impairment in photosynthetic swing and metabolism is recognized in plants exposed to the consequences of ozone (79). Sulfur and nitrogen oxides are included in the production of acid precipitation and are harmful to plants and marine organisms

5. Related Work

According to characteristics of the prediction methods used in relevant studies [62], air pollutant concentration prediction methods can be divided into conventional methods with non_deep learning and those based on deep learning.

Combined with methods of predicting pollutant concen_trations in meteorology, environmental science, mathematics, and computer science, the conventional prediction methods can be further divided into four types, predictions of empiri_cal models based on historical data and statistical methods, predictions of probability models based on statistical and mathematical methods or models, predictions based on synthetic methods, and prediction models based on conventional machine learning.

In the past, most forecasting schemes used statistical analyzes or methods developed in atmospheric sciences; however, machine learning has come to the fore, especially in forecasting air pollution, due to atmospheric science's heavy reliance on domain expertise and large datasets. Statistical methods are easily applied to the numerical prediction of air pollution without the need for esoteric professional knowledge; however, this approach is severely limited by the nature of the data, which means that it does not apply to situations that have not yet occurred. The complexity of fluctuations in air pollution levels cannot be adequately addressed using historical data. Thus, there is significant room for advancement in the forecast of PM2.5 in the real world.

Empirical models [63] do not analyze the process but count correlation data and determine the link between parameters and variables to obtain the corresponding relationship. For instance, the relationship between monthly mean pollutant concentration and other pollutant concentrations is established using an empirical statistical method [64]. His_torical data of pollutant concentrations can be modeled to predict changes in concentration through a chemical con_version model , [65]. Probability models are based on statistical probability regularity and are combined with statistical or mathematical modeling methods. They are used to produce or select more precise prediction samples. Such research is based on experiments, and its predictions are established on probability and statistical models. Dong et al. used a hidden semi-Markov model and added temporal structures. Past meteorological measurements and the past historical observation concentration level of PM2.5 were added to the training dataset, and corresponding Hidden Semi-Markov models (HSMMs) were trained for each concentration level. Prediction accuracy exceeded 24 hours.

Balachandran et al. used Bayesian algorithms to investigate the effects of various pollutant sources on their concentrations [66]. Using the ensemble Kalman filter method to construct a regional pollution assimilation system is a classic synthetic approach that combines numerical models with observations using the optimal estimation method. Backpropagation (BP) neural networks are frequently used in predictions based on conventional machine learning. Reference [67] considered Shenyang City as the monitoring center of a dataset as original data. The 120 sets of data in the 1999 and NOx concentration data were selected as the training set, and mete_orological data from 2000 were used as the test set. The prediction model of pollutant concentration was established [68], and the predicted results were obtained and compared with observations.

The above methods effectively predict small-scale air pollutant concentration. However, large amounts of background data and the daily accumulation of air pollution-related data are not independent. They have a time-dependent and spatial correlation. Conventional machine learning models do not have a deep network layer to limit high training costs and no coupling between the same layer of neurons, so they cannot solve time-dependent pollutant concentration.

Recently, the academic community has begun using deep neural networks for pollutant concentration prediction because of the shortcomings of conventional prediction meth_ods. Kuramoto et al. [69] used a deep network composed of two restricted Boltzmann machines to perform time-series prediction. Using the CATs benchmark [70] and original data, it has been proven that RBMs are superior to the ARIMA linear model. Ong et al. [71] predicted air pollutant concentration with deep recurrent neural networks, which have been widely researched [72]. This shows that DRNN yields better results than RBMs under the same conditions.

However, massive input data should be processed and features and correlations extracted. Then, time-series features should be extracted because pollution constitutes dependent on past historical data. Owing to its unique structure, CNN can use convolution kernels to convolve features of neighboring regions to obtain the features' spatial correlation [73]. The extraction of spatial correlations helps determine the effects of air quality and meteorological conditions of neighboring cities on target cities. LSTM is superior in processing time-series data because the concentration of air pollutants is time-dependent, and the historical concentration affects the future concentration.

Combining CNN and LSTM as the prediction model. The CNN convolutional layer was used as the basis for extracting features, and its shareable local weights reduced network complexity. Compared with RNNs, LSTM can solve long-term dependence problems and predict pollutant concentration in a time series. Therefore, these two networks were combined and could extract features in both spatial and temporal dimensions. Further, spatial and temporal effects can be introduced in the prediction system to obtain better predictions.

5.1 **More related work by researchers**

- Sun used a hidden Markov model to predict air pollution values.
- Delavar made similar predictions using a support vector machine.

- Hu, Huang, Liu, and Stafoggia used the Random Forest algorithm to make numerical predictions of PM2.5 concentrations.
- Many researchers have started to use shallow neural networks or deep learning models for predicting air pollution values.
- Lagesse used a shallow neural network to predict indoor PM2.5 values.
- Wang used a basic neural network to predict air pollution in Chongqing, China.
- Liu and Chen designed a three-step hybrid neural network model to help numerically predict outdoor PM2.5 levels.
- Pak used spatiotemporal correlations in conjunction with a deep learning model to predict air pollution values in Beijing.
- Liu used short-term, long-term memory based on attention and ensemble learning to predict air pollution concentrations.
- Zheng used micro-satellite images in conjunction with a convolutional neural network and a random forest approach to estimate PM2.5 values at the surface.
- Chen and Li designed a radial basis function - a long short-term memory to identify key weather observations at nearby stations to help predict PM2.5 values.

5.2 Table of previous works

The Pollutant	**City**	**Reference**	**Dataset Date**	**Deep Learning Models**
$PM_{2.5}$	Beijing city	Xiang Li	2014-2016	LTSM
PM10	Mexico	Ramírez Montañez	2002-2008	LTSM
PM10, PM2.5	Constantine	Ahmed Terrouche	23/03/2011-22/11/2011	

		Hocine Ali-Khodja		
PM_{10}, PM2.5, NO, NO2, Nox, NH3,	Visakhpatnam	K Rao Srinivasa	7/2016-5/2018	LSTM based RNN
$PM_{2.5}$	Taiwan	(Yi-Ting Tsai and al. 2018)	2012 – 2016	Le RNN with LSTM
$PM_{2.5}$	South korea (Séoul)	(Bui, T. C and al. 2018)	07/2008 – 04/2018	Encoder-Decoder Networks & LSTM
SO2, NOx, O3, CO, PM2.5,	India	S Geetha, L Prasika	1/2015-7/2018	LSTM
PM2.5, PM10, SO2, NO2,	China	Zepeng Qin	1/11/2018-20/11/2018	KNN-LSTM
PM10 PM2 .5	Daejeon, Gwangju, Daegu, Ulsan, and Busan	H. S. Kim	1/2014-5/2016	LSTM-based prediction
PM10, PM2.5, NO2, SO2, O3, CO	Shanghai	YongMing Pan	2014-2016	LSTM Wavelet-LSTM
CO, NO 2, O3, PM10, SO 2	Madrid	R. Navares	1/2001-12/2013	IGP-LSTM FC-LSTM SP-LSTM
PM10	CHINA Gobi	Jin, J., Lin, H.	1/2013-5/2015	CTM LOTOS-EUROS

6. Conclusion

In this chapter we saw an introductions to the environment issue that we studying that is climate pollution , then we talked about our main pollutant that we trying to predict in our work and that's the PM2.5 and the health and environment outcomes behind it , and finally we mentiond many related works in this domain by in diffrent counties , different searchers and diffrent methods .

CHAPTER III

AREA AND DATASET

III AREA AND DATASET

1. Introduction

This chapter contains a general description of our studied area with some details about it's geographic and climatic state. Also, an overview of the dataset that we going to work with , which was collected from that area.

2. Presentation of the area

2.1 Presentation of the area

2.1.1 Site

The city of Algiers is located in the north-central of the Republic of Algeria, and overlooks the western side of the Mediterranean, and consists of two parts: a part known as the Kasbah and extends over the edge of a steep hill (122 meters above sea level), and a modern part located at the level of the coast near the sea.

The city expanded to the northwest at the foot of Mount "Bouzareah", which is 400 meters high, and extended towards the east behind the mouth of "Wadi El Harrach" at the expense of the fertile lands of the "Mitidja" plain, along towards the south and southwest on the sloping hills of the coast, where the city swallowed Former agricultural villages[80].

2.1.2 Population

According to the World Geographical Dictionary, the city of Algeria was classified among the 100 largest cities in the world in terms of population, with an estimated 5.3 million people, and it is the largest city in the Maghreb region by population.

Its population consists of a group of Arabic speakers, Berbers and other ethnicities, including French and sub-Saharan Africans[80].

2.1.3 Economy

The Algerian capital monopolizes the main role in the economic, commercial and financial activity of the country, as the administrative center of Algeria it supervises most of the vital sectors, thanks to its port, which covers 4% of the movement of economic maritime activity in Algeria, and is considered a shipping center and a main refueling station in the Mediterranean Sea. .

Many industries are concentrated in the capital, especially in the Rouiba Raghaya region, on an area of 1079 hectares, and it has 30 major industrial units, the most important of which are bus and truck factories, industrial vehicles, chemical industries, cement, food industries, and the manufacture of clothes and shoes[80].

2.1.4 Climate of the state of Algiers

The climate of the wilaya of Algiers is average relative to the neighboring inland regions that reach huge daily temperature ranges, according to the Köppen climate classification, and the average precipitation ranges around 600 millimeters, which is equivalent to 24 inches annually, and the bulk of it is between monthly October and April, as for snowfall, is very rare in the state. For the year 2012, the state witnessed an accumulation of 100 millimeters, equivalent to 4 inches, and that was after an interruption of eight years. [81]

Figure 6:Map of Algeria Capital

2.2 **<u>Studied area and air pollution</u>**

2.2.1 <u>Pollution from transportation:</u>

the frequent use of transportation means that lead to a concentration of lead In the air, for example, in Algiers and its environs, the presence of lead is twice its value Inference, according to the World Health Organization report for the year 85, includes the garage of cars 80% of cars have more than 20 years of driving time, resulting in a high percentage of emissions Of the toxic gases that are due to the poor quality of car engines and the type of benzene it contains A large amount of lead

2.2.2 Industrial pollution:

Due to the smoke and toxic waste that the factories leave behind In the open air, without taking into account the scientific methods of storage, and it is one of the essential industrial units Causing air pollution in Algeria:

Cement factories: which are an essential source of gas flow, with an annual flow of 4,596 Tons of nitrogen oxide, 12,000 tons of carbon oxide, and 1020,000 tons of Sulfur oxide.

Gypsum and lime production units: where the gypsum production unit flows to Woods, and one unit Lime Augran production is about 250.20 tons of fines per year and 70 tons of nitrogen oxide. And 20 tons of carbon oxide(2).

Refining factories: The most critical radiation emanating from these factories is the result of the combustion of gases ncinerators that contributed to the increase in the greenhouse gases(3)

3. DataSet

Our dataset is a time series that contain two columns, one "date" contains the dates in M/D/Y formate, and the second one "pm25" represents the concentration of our pollutant in the air.

	A	B
1	date	pm25
2	5/1/2021	58
3	5/2/2021	62
4	5/3/2021	64
5	5/4/2021	63
6	5/5/2021	58
7	5/6/2021	57
8	5/7/2021	63
9	5/8/2021	67
10	5/9/2021	66
11	5/10/2021	57
12	5/11/2021	52
13	5/12/2021	55

3.1 **What is a time series data ?**

Time series analysis applies to the study of change in the data trend over a while. Time series analysis has a mixture of applications. One such application is the forecast of the future value of an object based on its past importance.

Those type of data are sequentially ordered data over time and these observations are typically collected at **regular intervals,** this could be:

- ***Every Second/Minute/Hour***
- ***Daily***
- ***Monthly***
- ***Quarterly/Yearly***

3.2 **Properties Of a Time Series Data**

- **Trends** show the overall tendency of the **data** to increase or decrease during an extended period. Generally, a **trend** is a smooth, broad, long-term, average tendency.

- **Seasonality:** refers to periodic changes. For example, electricity consumption is high throughout the day and low during the night; similarly, online sales go up during Christmas before going down again.

- **Stationarity** is a vital characteristic of the time series. A time series is stationary if its statistical characteristics do not change over time. In other words, it has **fixed mean and variance**, and covariance is independent of time. i.e., A Stock price is not a stationary series since we might see a growing or decreasing trend, and its volatility might increase over time(meaning that variance is changing).

4. Treating the Dataset

4.1 **Discription and Indexing**

date	pm25
2021-05-01	58
2021-05-02	62
2021-05-03	64
2021-05-04	63
2021-05-05	58

Figure 8:Representaion of the dataset

	pm25
count	669.000000
mean	67.789238
std	15.285765
min	40.000000
25%	58.000000
50%	65.000000
75%	74.000000
max	172.000000

Figure 7:Description of the dataset

4.2 **Visualization**

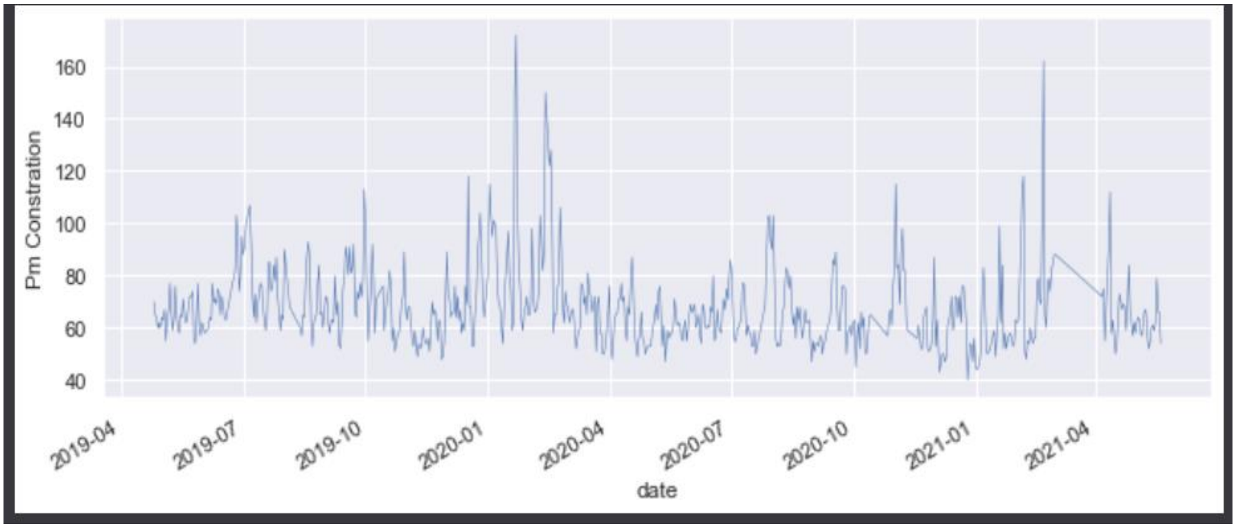


Figure 9:Plot represent PM constration

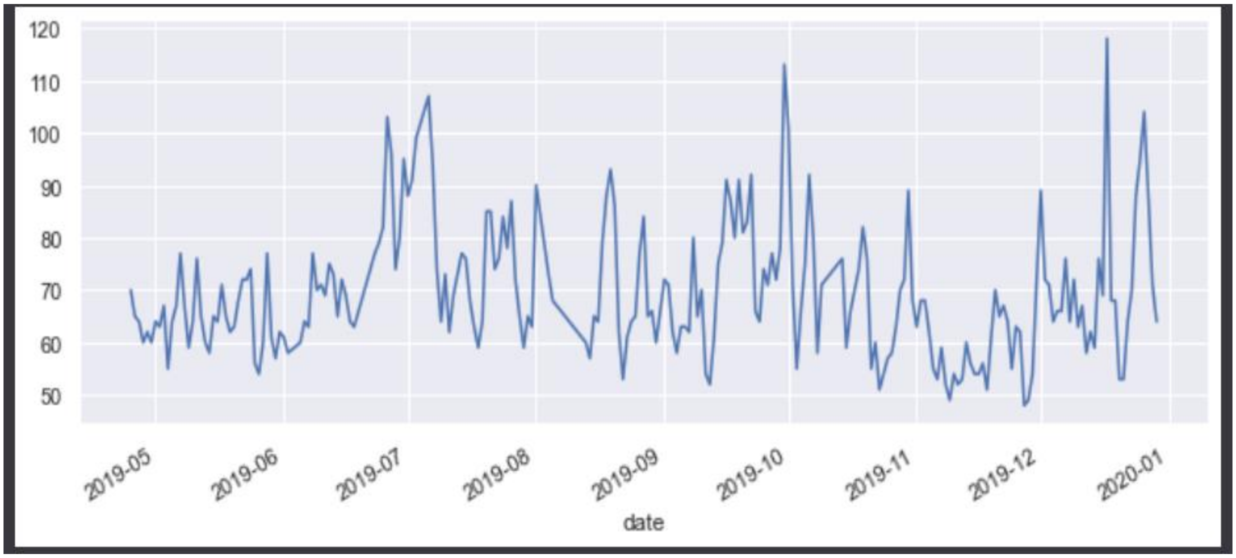


Figure 10:Plot represent PM constration in 2019

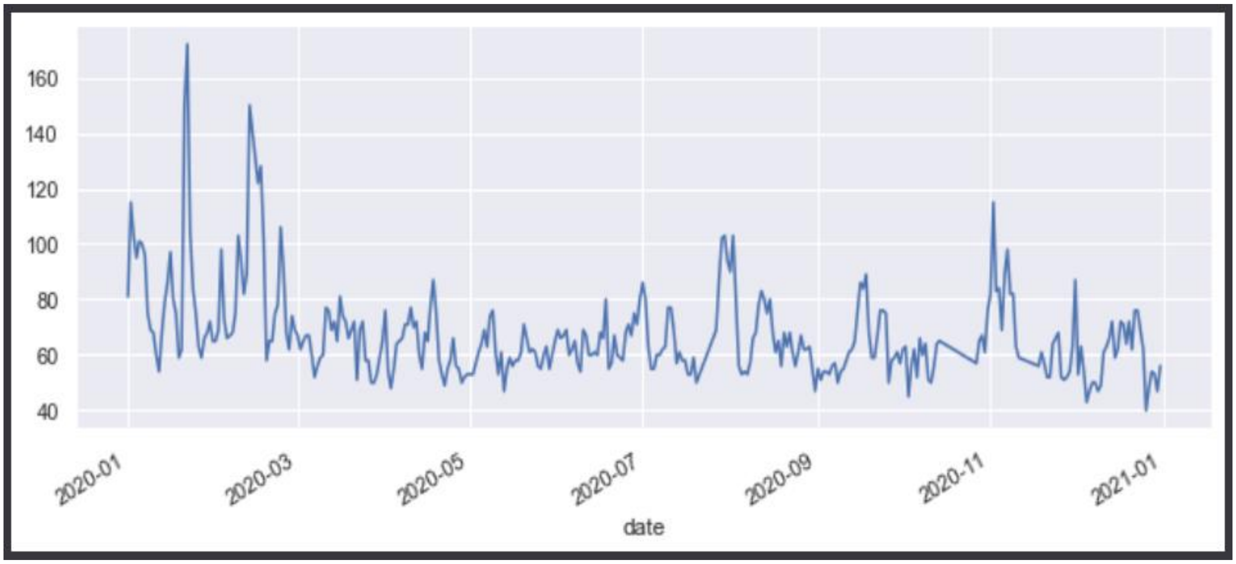


Figure 11:Plot represent PM constration in 2020

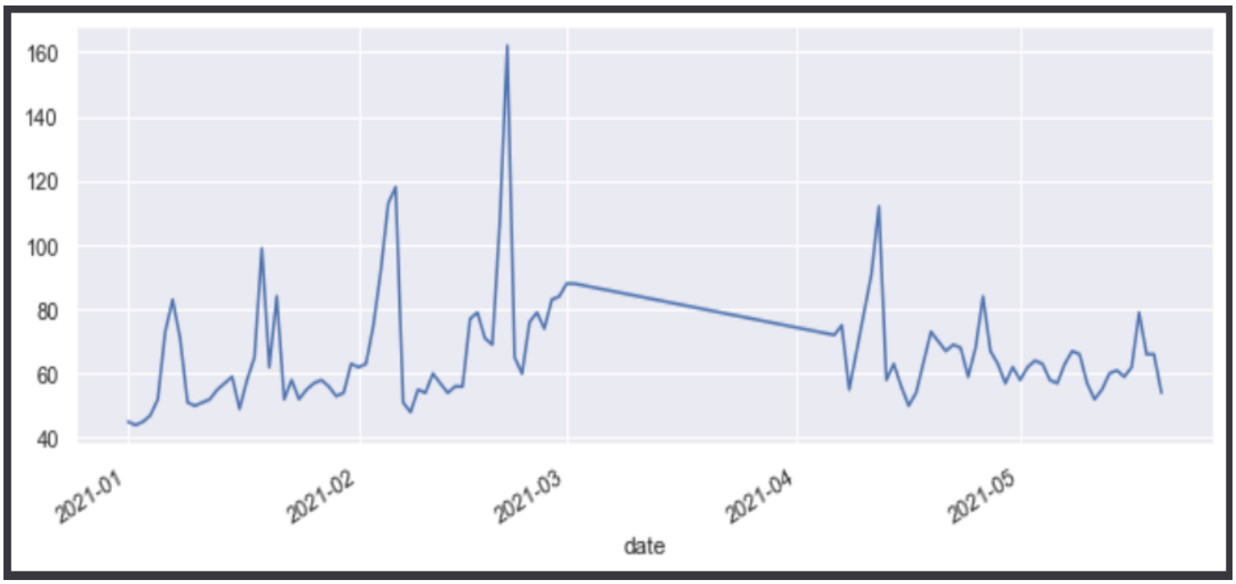


Figure 12:Plot represent PM constration in 2021

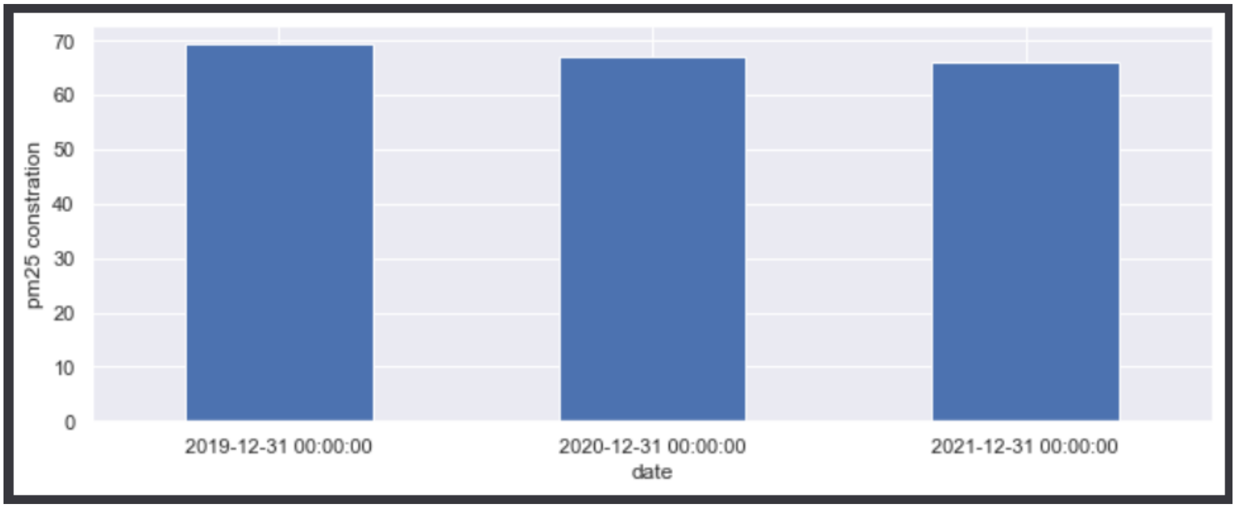


Figure 13:Plot represent mean of PM constration over the years

4.3 <u>**Check Stationarity**</u>

Stationarity may be a property of a time series. A stationary series is one where the values of the series aren't a function of time.

The statistical characteristics of the series like mean, variance, and autocorrelation are constant over time.

ADF test is the most typically used test for the Stationarity of the series, where the null hypothesis is the time serie possesses a unit root and is non-stationary. So if the p-Value in the ADH test is a smaller amount than the importance level (0.05) we reject the null hypothesis.

KPSS((Kwiatkowski-Phillips-Schmidt-Shin) test is another test for checking the stationarity of a statistic.

```
ADF Statistic: -11.872999688535002
p-value: 6.43753230206811e-22
Series is Stationary
KPSS Test Statistics: 0.07732137068185424
p-value: 0.1
Series is Stationary
```

Figure 14:Stationarity check

4.4 **Autocorrelation**

Autocorrelation is important because it can help us discover patterns in our time series, successfully select the best modeling algorithm, accurately evaluate the effectiveness of our model. Specifically, autocorrelation and partial autocorrelation plots are heavily used to summarize the strength and relationship within observations in a time series with observations at prior time steps.

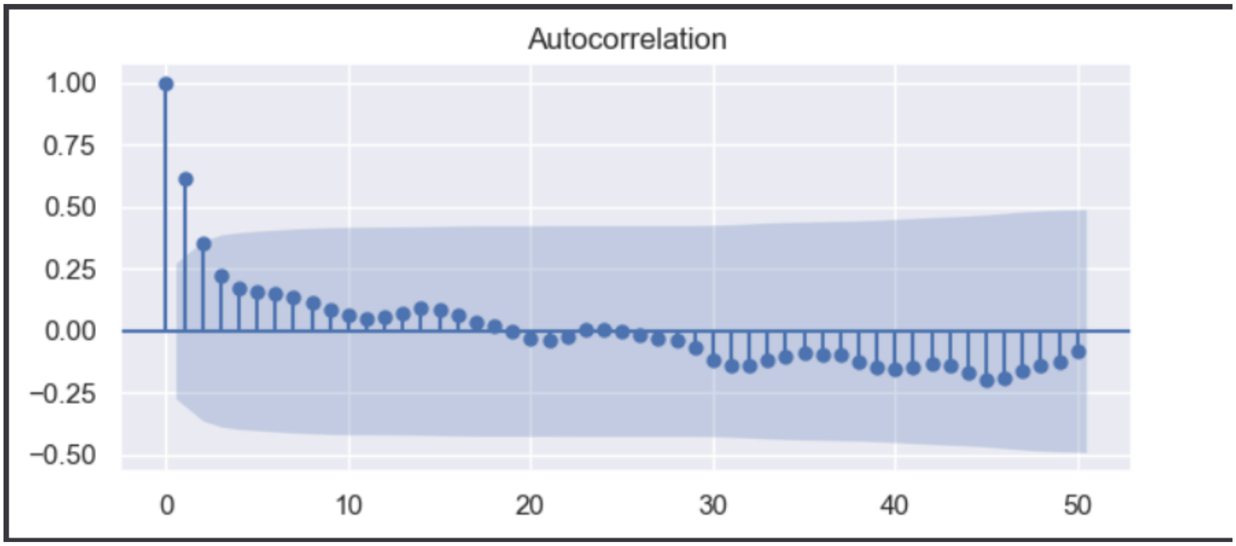


Figure 15:Plot represent Autocorrelation of the serie

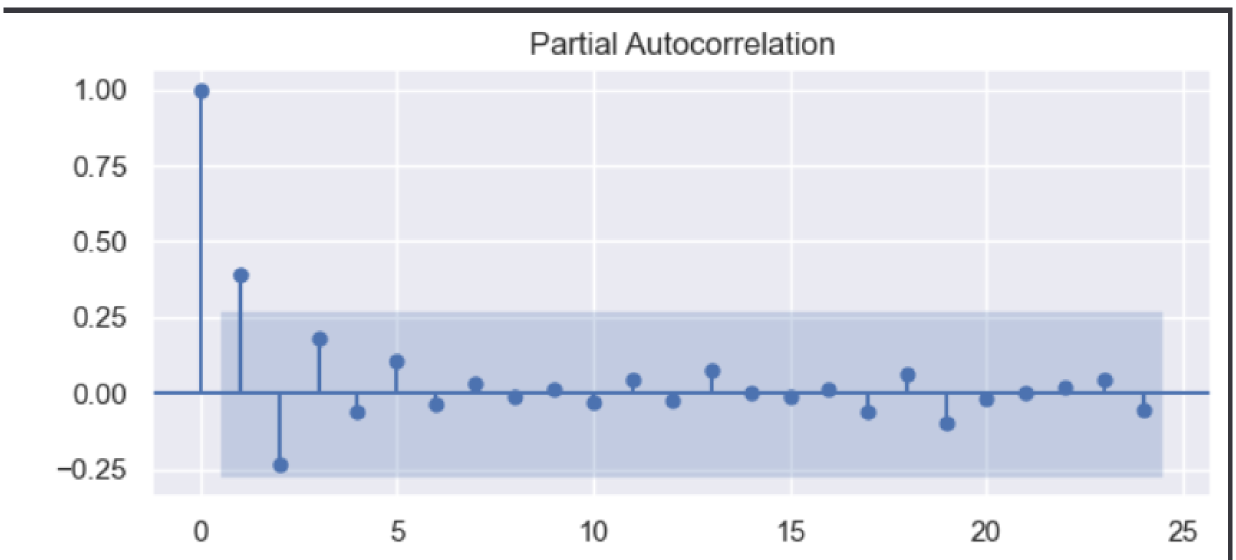


Figure 16:Plot represent Autocorrelation of the serie

4.5 <u>**Decomposition**</u>

Any time series may be split into the following components:

- **Base Level**
- **Trend**
- **Seasonality**
- **Error**

However, It is not mandatory that all-time series must have a trend and/or seasonality. A time series may not have a distinct trend but have a seasonality. The opposite can also be true.[1]

Depending on the nature of the trend and seasonality, a time series can be modelled as an **additive or multiplicative**, wherein, each observation in the series can be expressed as either a sum or a product of the components:

	Seasonality	Trend	Residual	Actual_values
date				
2019-04-25	0.0	70.0	0.0	70.0
2019-04-26	0.0	65.0	0.0	65.0
2019-04-27	0.0	64.0	0.0	64.0
2019-04-28	0.0	60.0	0.0	60.0
2019-04-29	0.0	62.0	0.0	62.0

Figure 17:Decomposition of the dataserie

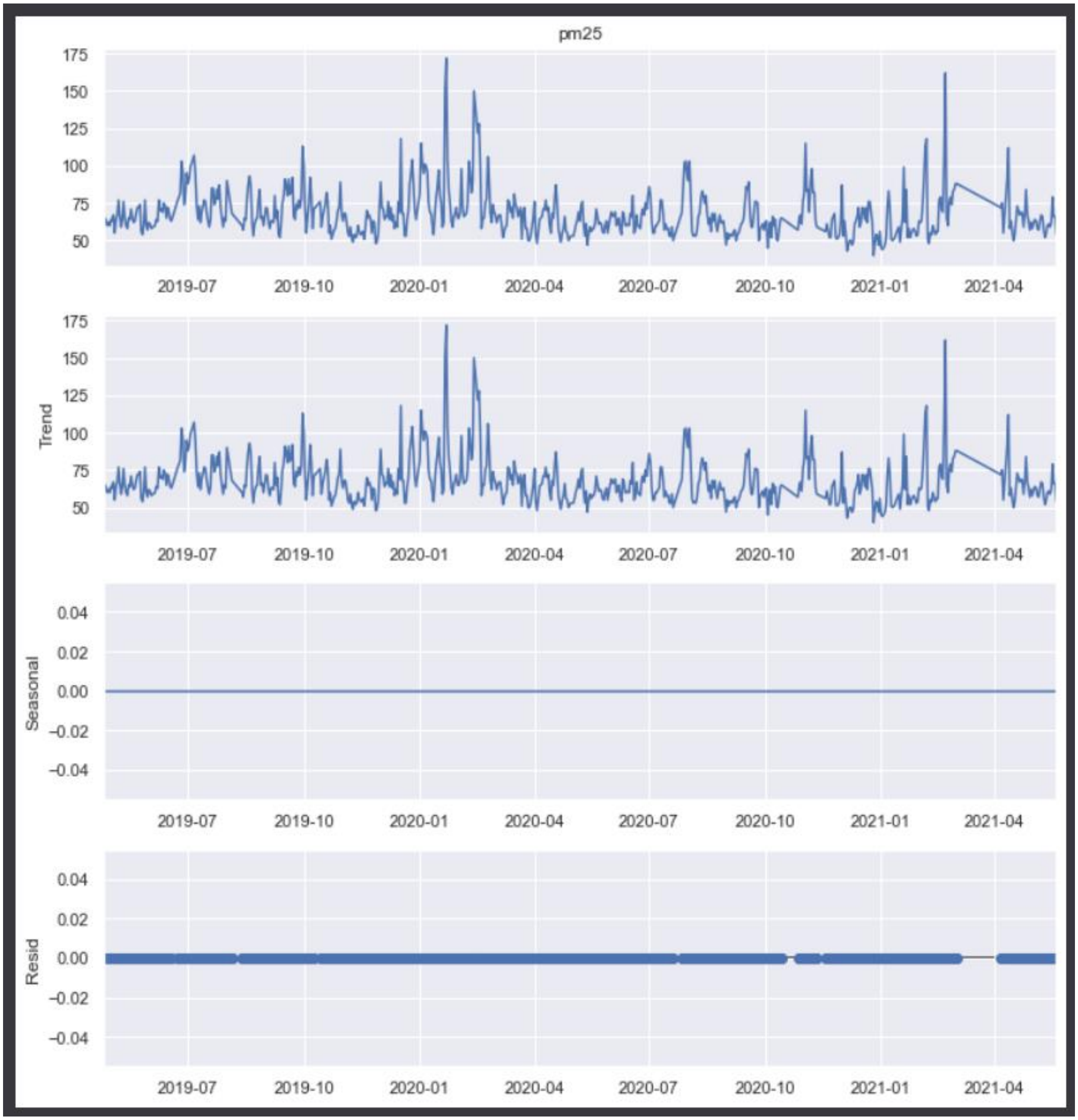


Figure 18:Fig represents the Trend Seasonality of the dataserie

4.6 **Resampling**

The df.resample() function let us resample our time series to our desired frequencies: A number of string aliases are given to use common time series frequencies, these are referred as *offset aliases*.

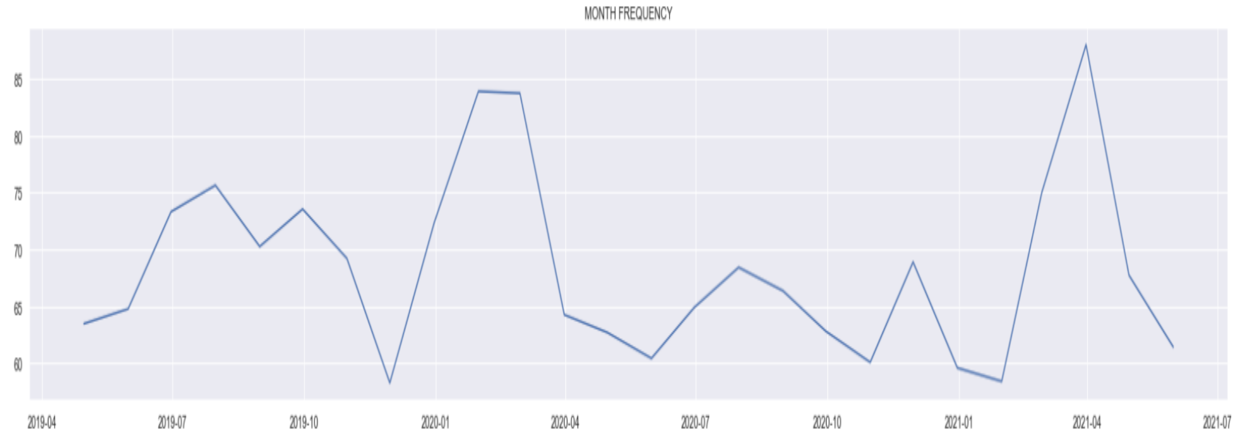


Figure 19:Plot represent Monthly Frequency of the dataserie

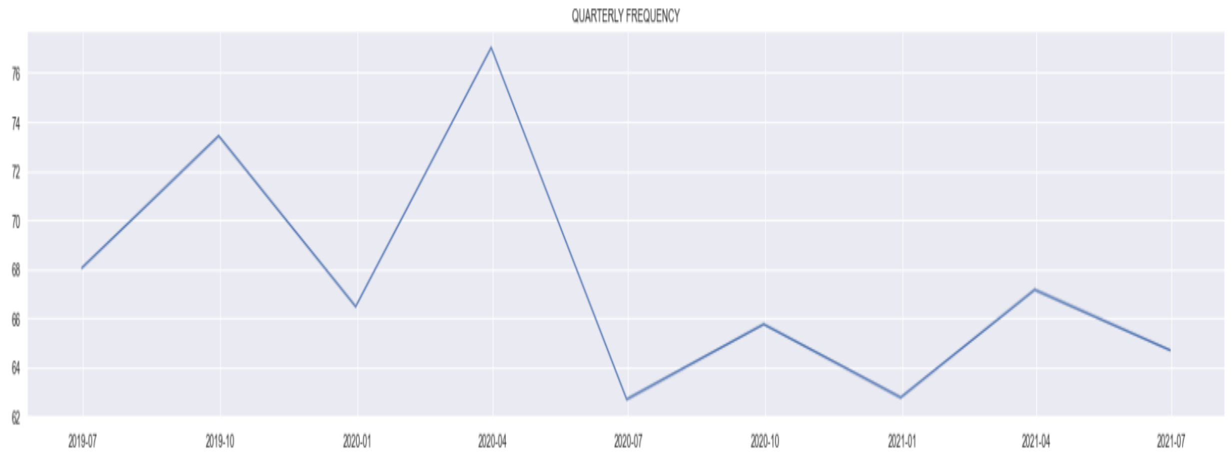


Figure 20:Plot represent Quarterly Frequency of the dataserie

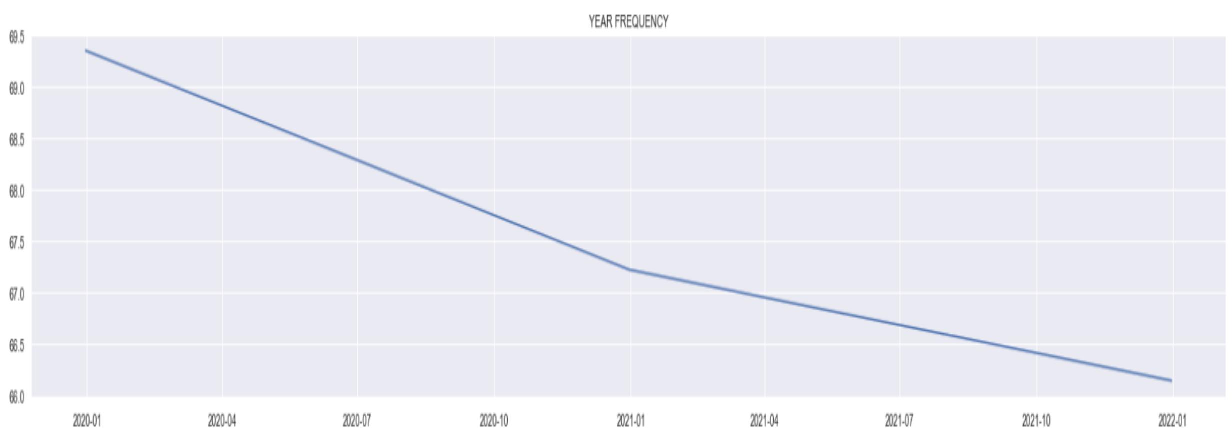


Figure 21:Plot represent Yearly Frequency of the dataserie

4.7 **Groupby**

Groupby() calculate the average mean over all the months, years and day of the week. We will apply **pct_change** to see how the pattern changed compared to the previous.

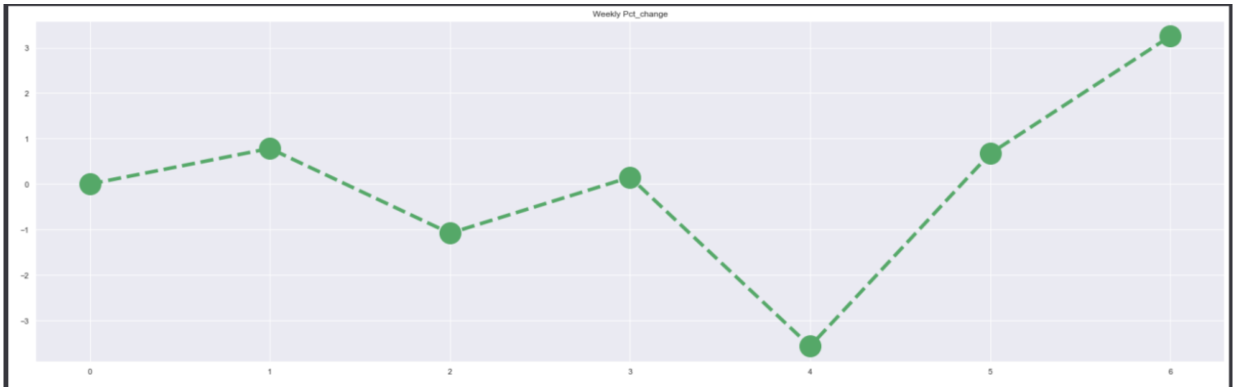


Figure 22:Plot represent percentage change over the weeks

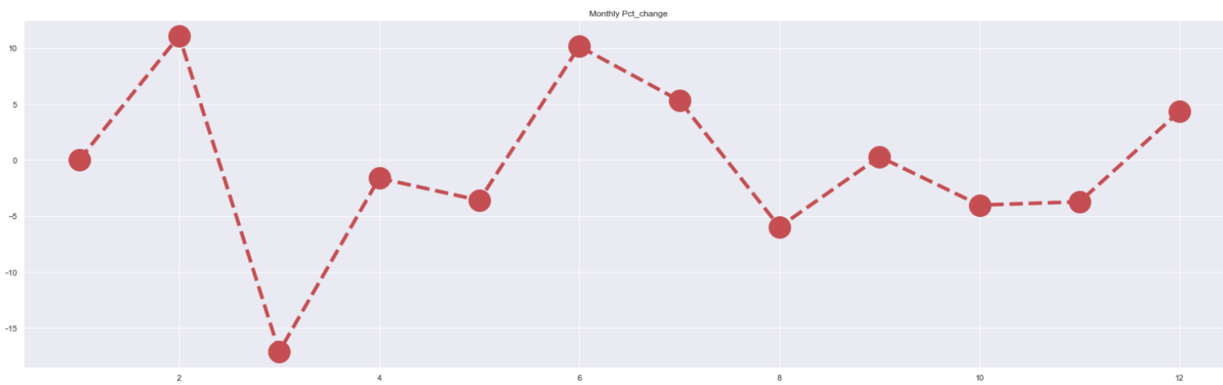


Figure 23:Plot represent percentage change over the months

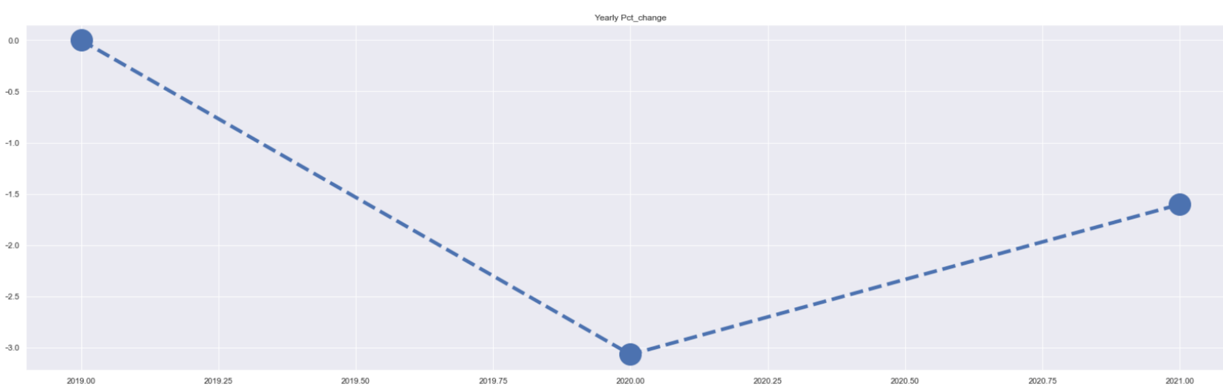


Figure 24:Plot represent percentage change over the year

CHAPTER IV

MODULES AND RESULTS

IV MODULES AND RESULTS

1. Introduction

In this chapter we will discuss the experiments and the implementation of our model using our dateset, as well as all the steps followed for its training, also we present all the development environment and the tools used in this project for creating the prediction model as long as the libraries we needed, finally the results will be explained and discussed.

2. Development environment

2.1 Tools

2.1.1 Visual studio

Visual Studio .NET is Microsoft's visual programming environment for creating Web services based on use of the Extensible Markup Language (XML). The product suite provides a visual interface for identifying a program as a Web service, forms for building a user interface (including support for mobile device interfaces), features for integrating existing application data, and for debugging. Visual Studio .NET comes with the .NET Framework, including the Common Language Runtime, and includes several programming languages including Visual Basic, Visual C++, and Visual C# [82].

2.1.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed [83].

2.1.3 Jupyter

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones [84].

2.2 **Libraries**

Library	**Description**
TensorFlow	Primarily used for developing and training highly efficient Machine Learning and Deep Learning models, TensorFlow can also help you deploy these models to a host of platforms, such as a CPU, GPU, or a TPU(Tensor Processing Unit), with ease.

Pandas	Pandas is another popular high-performance Python library that is being widely used today for solving modern Data Science and Machine Learning problems. By offering developers access to flexible yet extremely responsive data structures for working with time series and structured data along with the stack of other vital features, Pandas aims to become the best data analysis tool available for solving real-world problems. A brief rundown of the features offered by Pandas include: ● An efficient DataFrame object for data manipulation ● Easy reshaping and pivoting of data sets ● Merging and joining of data sets ● Label-based data slicing, indexing, and subsetting ● Allows working with time-series data
Numpy	NumPy is one of the most used libraries for tasks involving modern scientific computations and evolving yet powerful domains like Data Science and Machine Learning. NumPy also plays well with many other visualization libraries, such as **Matplotlib, Seaborn, and Plotly**.
Keras	Keras is designed to be simple while reducing the cognitive load, leading to lesser user interaction in most use cases. This feature makes Keras one of the coolest Python libraries for learning about Deep Neural Networks. Keras offers plenty of firepower to help you quickly create model prototypes. Keras' impressive deployment capabilities let you deploy your models on a variety of platforms, including web browsers, embedded devices, iOS, and Android devices.
Matplotlib	Matplotlib is undoubtedly one of the most popular visualization libraries for Python. Being used by hundreds of companies and individuals, matplotlib lets you visualize your data in several different ways. You can use it to create a wide variety of visualizations, including line plots, histograms, bar charts, pie charts, scatter plots, tables, and many other styles.

ARIMA	ARIMA, short for 'Auto Regressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values. Any 'non-seasonal' time series that exhibits patterns and is not a random white noise can be modeled with ARIMA models.

3. The Creation of our LSTM network

We can handle the problem as a regression problem

That is, given the Pm consternation (in units) this day, what is the pm concentration the next day?

We write a simple function to transform our single column of data into a two-column dataset: the first column containing this day (t) PM count and the second column containing next day (t+1) PM count, to be predicted.

Let's first import all of the functions we aim to use. This implies a working SciPy environment with the Keras deep learning library installed.

```
import numpy
import matplotlib.pyplot as plt
from pandas import read_csv
import math
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
```

Before starting, we can fix the random number seed to guarantee our results are reproducible.

```
numpy.random.seed(7)
```

Next we load the dataset as a Pandas data frame. Then we extract the NumPy array from the data frame and convert the integer values to floating-point values, which is more suitable for modeling with a neural network.

```
dataframe = read_csv('C:/Users/Djame/Desktop/Project/AlgeriaDataSet2.0.csv', usecols=[1], engine='python')
dataset = dataframe.values
dataset = dataset.astype('float32')
```

LSTMs are sensitive to the measure of the input data, specifically if the sigmoid (default) or the activation functions are used. It can be a good practice to rescale the data to the range of 0-to-1, also called normalizing. We can normalize the dataset applying the MinMaxScaler preprocessing class from the scikit-learn library.

```
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset)
```

After we model our data and evaluate the skill of our model on the training dataset, we need to know the skill of the model on new hidden data.

With time-series data, the sequence of values is important. A simple method that we can use is to split the required dataset into train and test datasets. The code below calculates the index of the division point and separates the data into the training datasets with 67% of the observations that we can use to train our model, leaving the unused 33% for testing the model.

```
train_size = int(len(dataset) * 0.67)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
```

Now we define a function to create a new dataset.

The function takes two arguments: the **dataset**, which is a NumPy array that we want to transform into a dataset, and the **look_back**, which is the number of previous time steps to use as data variables to predict the next time period — in this case defaulted to 1.

This default will generate a dataset where X is the PM construction at a given time (t) and Y is the PM construction at the next time (t + 1).

```
def create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back-1):
        a = dataset[i:(i+look_back), 0]
        dataX.append(a)
        dataY.append(dataset[i + look_back, 0])
    return numpy.array(dataX), numpy.array(dataY)
```

Now we use this function to prepare the train and test datasets for modeling.

```
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
```

The LSTM network requires the input data (X) to be provided with a precise array structure in the form of [samples, time steps, features].

Our data is in the form: [samples, features] and we are planting the problem as one time step for each sample. We can transform the prepared train and test input data into the predicted structure using **NumPy.reshape()** :

```
trainX = numpy.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = numpy.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
```

Now we ready to create and fit our LSTM network for this problem.

The network has a visible layer with 1 input, a hidden layer with 4 LSTM blocks or neurons, and an output layer that makes a unique value prediction. The default sigmoid activation function is used for the LSTM blocks. The network is trained for 100 epochs and a batch size of 1 is used.

```
model = Sequential()
model.add(LSTM(4, input_shape=(1, look_back)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)
```

When the model is suitable, we can determine the performance of the model on the train and test datasets. This will give us a point of comparison for new models.

Note that we invert the predictions before calculating error scores to assure that performance is reported in the same units as the original data.

```
# make predictions

trainPredict = model.predict(trainX)
testPredict = model.predict(testX)

# invert predictions

trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])

# calculate error

trainScore = math.sqrt(mean_squared_error(trainY[0], trainPredict[:,0]))
print('Train Score: %.2f RMSE' % (trainScore))
testScore = math.sqrt(mean_squared_error(testY[0], testPredict[:,0]))
print('Test Score: %.2f RMSE' % (testScore))
```

Finally, we can make predictions using the model for the train and test dataset to get a visual implication of the skill of the model.

Next, we must shift the predictions so that they follow on the x-axis with the primary dataset. Once fixed, the data is planned, showing the primary dataset in blue, the predictions for the training dataset in orange, and the predictions on the unseen test dataset in green.

```
# shift train predictions for plotting

trainPredictPlot = numpy.empty_like(dataset)
trainPredictPlot[:, :] = numpy.nan
trainPredictPlot[look_back:len(trainPredict)+look_back, :] =

# shift test predictions for plotting

testPredictPlot = numpy.empty_like(dataset)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, :] = testPredict

# plot baseline and predictions

plt.plot(scaler.inverse_transform(dataset))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
```

4. Results of multiple LSTM modules :

LSTM MODULE	Lookback = 1	Lookback = 7	Lookback = 14	Lookback = 30	**Best Result**
LSTM Network for Regression	Train Score : 11,16 Test Score : 12,65	Train Score : 11,14 Test Score : 12,73	Train Score : 11,16 Test Score : 12,85	Train Score : 11,14 Test Score : 12,73	Lookback 1
LSTM for Regression Using the Window Method	Train Score : 11,17 Test Score : 13,15	Train Score : 11,00 Test Score : 12,78	Train Score : 11,04 Test Score : 12,97	Train Score : 10,98 Test Score : 12,75	Lookback 30
LSTM for Regression with Time Steps	Train Score : 11,02 Test Score : 12,70	Train Score : 11,05 Test Score : 13,05	Train Score : 10,98 Test Score : 12,64	Train Score : 11,03 Test Score : 13,04	Lookback 14
LSTM with Memory Between Batches	Train Score : 11,04 Test Score : 12,87	Train Score : 11,04 Test Score : 12,65	Train Score : 11,01 Test Score : 12,73	Train Score : 10,99 Test Score : 12,62	Lookback 30
Stacked LSTMs with Memory Between Batches	Train Score : 11,04 Test Score : 12,56	Train Score : 11,02 Test Score : 12,66	Train Score : 11,09 Test Score : 12,72	Train Score : 10,85 Test Score : 12,67	Lookback 1

- Ploting the best result for the Stacked LSTM with Memory Between Batches

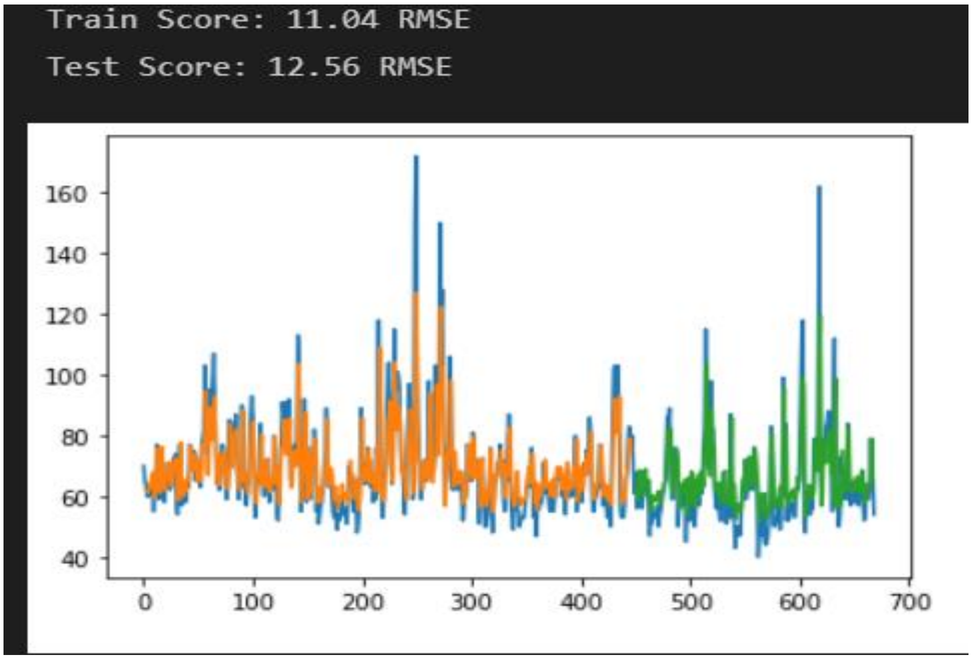


Figure 25: Plot represent the results of the LSTM with memory between batches module

We can notice that the model did a great job fitting both the training and the test datasets.

5. Conclusion

After testing many LSTM modules in our data set and changing our lookback in 4 different ways (daily, weekly, half monthly, and monthly) and since we are only interested in the Test-score because that's where our hiding data being predicted, we managed to get the best results possible from our table above and that's when we used the monthly and daily lookback since they gave us the lowest error test score possible in 2 different algorithms, and from all our 5 methods we conclude that the Stacked LSTM with Memory Between Batches is the best algorithm applied to our data set with a lookback of 1day it gave us an error test score of 12,56.

Conclusion

Air pollution peak’s period is a real public health crisis, to help the managers and the decision-makers to manage the situation. Prediction model can help by providing forecasting of the concertation of air pollutant. Therefore, having this crucial information before the happening of the crisis is a key element. Public authority can respond, by limiting transportation traffic, polluting activities and informing vulnerable population.

In this work, multiple architectures of LSTM model are presented, each one showed different performances. The model are trained using real data from air pollution’s control station about Algiers. Our model have showed an interesting performances specifically when we use back window of size of multiple of 7 days. It confirms that the air pollution has a weekly seasonality, and thus, including this information in future model could lead to increasing the accuracy.

References :

[1] United Nations, Department of Economic and Social Affairs, Population Division. World Population Prospects: The 2015 Revision, Key Findings and Advance Tables. 2015. Working Paper No. ESA/P/WP.241. Retrieved from: https://esa.un.org/unpd/wpp/ publications/files/key_findings_wpp_2015.pdf

[2] World Health Organization, Media Centre. Air pollution levels rising in many of the world's poorest cities [Internet]. 2016. Available from: http://www.who.int/mediacentre/ news/releases/2016/air-pollution-rising

[3] UNEP. Status of Fuel Quality and Vehicle Emission Standards Latin America [Internet]. 2016. Available from: http://www.unep.org/urban_environment/Issues/urban_air. [Accessed: 22 October 2016]

[4] Lelieveld J, Evans JS, Fnais M, Giannadaki D, Pozzer A. The contribution of outdoor air pollution sources to premature mortality on a global scale. Nature. 2015;525(7569): 367-371

[5] Karagulian F, Belis CA, Dora CFC, Prüss-Ustün AM, Bonjour S, Adair-Rohani H, Amann M. Contributions to cities' ambient particulate matter (PM): A systematic review of local source contributions at global level. Atmospheric Environment. 2015;120:475-483

[6] Castell N, Dauge FR, Schneider P, Vogt M, Lerner U, Fishbain B, Broday D, Bartonova A. Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? Environment International. 2017;99:293-302

[7] Kleine Deters J, Zalakeviciute R, Gonzalez M, Rybarczyk Y. Modeling PM2.5 urban pollution using machine learning and selected meteorological parameters. Journal of Electrical and Computer Engineering. 2017. 14 pages. Article ID 5106045. DOI: 10.1155/2017/5106045

[8] Marshland, S. (2009) Machine Learning an Algorithmic Perspective. CRC Press, New Zealand, 6-7.

[9] https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[10] Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. Nature 1986, 323, 533–536. [CrossRef].

[11] Elman, J.L. Finding structure in time. Cogn. Sci. 1990, 14, 179–211. [CrossRef].

[12] Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. 1994, 5, 157–166. [CrossRef] [PubMed]

[12] Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 1998, 6, 107–116. [CrossRef]

[13] Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. III-1310–III-1318

[14] Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780. [CrossRef]

[15] Olah, C. Understanding LSTM Networks. Available online: http://colah.github.io/posts/2015-08- Understanding-LSTMs/ (accessed on 26 June 2018).

[17] Felix A. Gers and J¨urgen Schmidhuber. LSTM recurrent networks learn simple context-free and context-sensitive languages. IEEE Trans. on Neural Networks, 12(6):1333–1340, jan 2001.

[16] Sepp; Hochreiter and J¨urgen Schmidhuber. Long Short-Term Memory. Neural computation, 9(8):1735–1780, 1997

[18] Felix A. Gers, Nicol N. Schraudolph, and J¨urgen Schmidhuber. Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research (JMLR), 3(1):115–143, 2002.

[19] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In Georg Dorffner, Horst Bischof, and Kurt Hornik, editors, Proc. of the Int. Conf. of Artificial Neural Networks (ICANN 2001), pages 87–94, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[20] Bram Bakker. Reinforcement learning with long short-term memory. In Advances in Neural Information Processing Systems (NIPS'02), 2002.

[21] Martin W¨ollmer, Florian Eyben, Stephan Reiter, Bj¨orn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie. Abandoning emotion classes - Towards continuous emotion recognition with modelling of long-range dependencies. In Proc. of the Ann. Conf. of the Int. Speech Communication Association (INTERSPEECH'08), number January, pages 597–600, 2008.

[22] Ilya Sutskever, James Martens, and Geoffrey E. Hinton. Generating text with recurrent neural networks. In Proc. of the 28th Int. Conf. on Machine Learning (ICML-11)., pages 1017–1024, 2011.

[23] Alex Graves. Generating sequences with recurrent neural networks. Proc. of the 23rd Int. Conf. on Information and Knowledge Management (CIKM '14), pages 101–110, aug 2014.

[24] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett, editors, Advances in Neural Information Processing Systems 28, pages 2773–2781. Curran Associates, Inc., dec 2014.

[25] Oriol Vinyals and Quoc V. Le. A neural conversational model. arXiv, 37, jun 2015.

[26] Yoshua Bengio, R´ejean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. The Journal of Machine Learning Research, 3:1137–1155, 2003.

[27] Herv´e A. Bourlard and Nelson Morgan. Connectionist Speech Recognition - a hybrid approach. Kluwer Academic Publishers, Boston, MA, 1994.

[28] N Beringer, A Graves, F Schiel, and J Schmidhuber. Classifying unprompted speech by retraining LSTM Nets. In W Duch, J Kacprzyk, E Oja, and S Zadrozny, editors, Artificial Neural Networks: Biological Inspirations (ICANN), volume 3696 LNCS, pages 575–581. Springer-Verlag Berlin Heidelberg, 2005.

[29] Santiago Fern´andez, Alex Graves, and J¨urgen Schmidhuber. Phoneme recognition in TIMIT with BLSTM-CTC. Arxiv preprint arXiv08043269, abs/0804.3:8, 2008.

[30]] Emanuel Indermuehle, Volkmar Frinken, Andreas Fischer, and Horst Bunke. Keyword spotting in online handwritten documents containing text and non-text using BLSTM neural networks. In Int. Conf. on Document Analysis and Recognition (ICDAR), pages 73–77. IEEE, 2011.

[32] Haim Sak, Andrew Senior, and Fran¸coise Beaufays. Long short-term memory based recurrent neural network architectures for large scale acoustic modeling. In Interspeech 2014, number September, pages 338–342, feb 2014.

[33] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence learning with neural networks. Advances in Neural Information Processing Systems (NIPS'14), pages 3104–3112, sep 2014.

[34] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, Attend and Spell. arXiv preprint, pages 1–16, aug 2015.

[35] Jan Koutnik, Klaus Greff, Faustino Gomez, and J¨urgen Schmidhuber. A Clockwork RNN. In Proc. of the 31st Int. Conf. on Machine Learning (ICML 2014), volume 32, pages 1863–1871, 2014

[36] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Proc. of the Int. Conf. on Learning Representations (ICLR 2015), volume 26, page 15, sep 2015

[37] Alex Graves, Marcus Liwicki, Santiago Fern´andez, Roman Bertolami, Horst Bunke, and J¨urgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. IEEE trans. on Pattern Analysis and Machine Intelligence, 31(5):855–68, may 2009.

[38] Thibauld Nion, Fares Menasri, Jerome Louradour, Cedric Sibade, Thomas Retornaz, Pierre Yves Metaireau, and Christopher Kermorvant. Handwritten information extraction from historical census documents. Proc. of the Int. Conf. on Document Analysis and Recognition (ICDAR 2013), pages 822–826, 2013.

[39] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. ArXiv e-prints, pages 1–18, 2012.

[40] Nal Kalchbrenner and Phil Blunsom. Recurrent Continuous Translation Models. In Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP'13), volume 3, page 413. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013.

[41] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Proc. of the Int. Conf. on Learning Representations (ICLR 2015), volume 26, page 15, sep 2015.

[42] Sebastian Otte, Dirk Krechel, Marcus Liwicki, and Andreas Dengel. Local Feature Based Online Mode Detection with Recurrent Neural Networks. Int. Conf. on Frontiers in Handwriting Recognition, pages 533–537, 2012.

[43] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. In Conf. on Computer Vision and Pattern Recognition (CVPR'15), pages 3156–3164, jun 2015.

[44] Front. Public Health, 20 February 2020 | https://doi.org/10.3389/fpubh.2020.00014

[45] Bezirtzoglou C, Dekas K, Charvalos E. Climate changes, environment and infection: facts, scenarios and growing awareness from the public health community within Europe. Anaerobe. (2011) 17:337–40. doi: 10.1016/j.anaerobe.2011.05.016

[46] Castelli F, Sulis G. Migration and infectious diseases. Clin Microbiol Infect. (2017) 23:283–9. doi: 10.1016/j.cmi.2017.03.012

[47] Watson JT, Gayer M, Connolly MA. Epidemics after natural disasters. Emerg Infect Dis. (2007) 13:1–5. doi: 10.3201/eid1301.060779

[48] Fenn B. Malnutrition in Humanitarian Emergencies. Available online at: https://www.who.int/diseasecontrol_emergencies/publications/idhe_2009_ london_malnutrition_fenn.pdf. (accessed August 15, 2017).

[49] Lindh E, Argentini C, Remoli ME, Fortuna C, Faggioni G, Benedetti E, et al. The Italian 2017 outbreak Chikungunya virus belongs to an emerging Aedes albopictus–adapted virus cluster introduced from the Indian subcontinent. Open Forum Infect Dis. (2019) 6:ofy321. doi: 10.1093/ofid/ofy321

[50] Calba C, Guerbois-Galla M, Franke F, Jeannin C, Auzet-Caillaud M, Grard G, Pigaglio L, Decoppet A, et al. Preliminary report of an autochthonous chikungunya outbreak in France, July to September 2017. Eur Surveill. (2017) 22:17-00647. doi: 10.2807/1560-7917.ES.2017.22.39.17-00647

[51] Menne B, Murray V. Floods in the WHO European Region: Health Effects and Their Prevention. Copenhagen: WHO; Weltgesundheits organisation, Regionalbüro für Europa (2013). Available online at: http://www.euro.who. int/data/assets/pdf_file/0020/189020/e96853.pdf (accessed 15 August 2017)

[52] Schneider SH. The greenhouse effect: science and policy. Science. (1989) 243:771–81. doi: 10.1126/science.243.4892.771

[53] Wilson WE, Suh HH. Fine particles and coarse particles: concentration relationships relevant to epidemiologic studies. J Air Waste Manag Assoc. (1997) 47:1238–49. doi: 10.1080/10473289.1997.10464074

[54] US EPA (US Environmental Protection Agency) (2018). Available online at: https://www.epa.gov/pm-pollution/particulate-matter-pm-basics (accessed September 22, 2018).

[55] Cheung K, Daher N, Kam W, Shafer MM, Ning Z, Schauer JJ, et al. Spatial and temporal variation of chemical composition and mass closure of ambient coarse particulate matter (PM10–2.5) in the Los Angeles area. Atmos Environ. (2011) 45:2651–62. doi: 10.1016/j.atmosenv.2011.02.066

[56] Zhang L, Yang Y, Li Y, Qian ZM, Xiao W, Wang X, et al. Shortterm and long-term effects of PM2.5 on acute nasopharyngitis in 10 communities of Guangdong, China. Sci Total Env. (2019) 688:136–42. doi: 10.1016/j.scitotenv.2019.05.470.

[57] Kloog I, Ridgway B, Koutrakis P, Coull BA, Schwartz JD. Long- and short-term exposure to PM2.5 and mortality using novel exposure models, Epidemiology. (2013) 24:555–61. doi: 10.1097/EDE.0b013e318294beaa

[58] New Hampshire Department of Environmental Services. Current and Forecasted Air Quality in New Hampshire. Environmental Fact Sheet (2019). Available online at: https://www.des.nh.gov/organization/commissioner/ pip/factsheets/ard/documents/ard-16.pdf (accessed September 22, 2019).

[59] Kappos AD, Bruckmann P, Eikmann T, Englert N, Heinrich U, Höppe P, et al. Health effects of particles in ambient air. Int J Hyg Environ Health. (2004) 207:399–407. doi: 10.1078/1438-4639-00306

[60] Boschi N (Ed.). Defining an educational framework for indoor air sciences education. In: Education and Training in Indoor Air Sciences. Luxembourg: Springer Science & Business Media (2012). 245 p

[61] Heal MR, Kumar P, Harrison RM. Particles, air quality, policy and health. Chem Soc Rev. (2012) 41:6606–30. doi: 10.1039/c2cs35076a

[62] A. Lee, A. Szpiro, S. Y. Kim, and L. Sheppard, ‘‘Impact of preferential sampling on exposure prediction and health effect inference in the context of air pollution epidemiology,’’ Environmetrics, vol. 26, no. 4, pp. 255–267, 2015.

[63] K. R. Felzer, ‘‘The empirical model,’’ Appendix Q Uniform Calfiornia Earthquake Rupture Forecast Version 3 (UCERF3), Time Indepdendent Model., USGS Open File Rep. 2013-1165, 2013.

[64] Y. H. Xie, M. M. Zhang, L. Yang, and H. D. Zhang, ‘‘Predicting urban PM2.5 concentration in China using support vector regression,’’ Comput. Eng. Des., vol. 11, pp. 3106–3111, 2015.

[65] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards real-time object detection with region proposal networks,’’ in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[66] I. Djalalova, L. D. Monache, and J. Wilczak, ‘‘PM2.5 analog forecast and Kalman filter post-processing for the Community Multiscale Air Quality (CMAQ) model,’’ Atmos. Environ., vol. 108, pp. 76–87, May 2015.

[67] S. Balachandran, H. H. Chang, J. E. Pachon, H. A. Holmes, J. A. Mulholland, and A. G. Russell, ‘‘Bayesian-based ensemble source apportionment of PM2.5,’’ Environ. Sci. Technol., vol. 47, no. 23, pp. 13511–13518, 2013.

[68] J. Wang, X. M. Hu, and L. X. Zheng, ‘‘Study on forecasting of air pollution based on BP model,’’ Res. Environ. Sci., vol. 15, no. 5, pp. 62–64, 2002.

[69] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, ‘‘Time series forecasting using a deep belief network with restricted Boltzmann machines,’’ Neurocomputing, vol. 137, pp. 47–56, Aug. 2014.

[70] A. Lendasse, E. Oja, O. Simula, and M. Verleysen, Time Series Prediction Competition: The CATS Benchmark. Amsterdam, The Netherlands: Elsevier, 2007.

[71] B. T. Ong, K. Sugiura, and K. Zettsu, ‘‘Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5,’’ Neural Comput. Appl., vol. 27, no. 6, pp. 1553–1566, 2016.

[72] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, ‘‘Advances in optimizing recurrent networks,’’ in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2013, pp. 8624–8628.

[73] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ‘‘Gradient-based learning applied to document recognition,’’ Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[74] Madronich S, de Gruijl F. Skin cancer and UV radiation. *Nature*. (1993) 366:23–9. 10.1038/366023a0 [PubMed] [CrossRef] [Google Scholar]

[75] Teramura A. Effects of UV-B radiation on the growth and yield of crop plants. *Physiol Plant*. (2006) 58:415–27. 10.1111/j.1399-3054.1983.tb04203.x [CrossRef] [Google Scholar]

[76] Singh E, Tiwari S, Agrawal M. Effects of elevated ozone on photosynthesis and stomatal conductance of two soybean varieties: a case study to assess impacts of one component of predicted global climate change. *Plant Biol Stuttg Ger*. (2009) 11(Suppl. 1):101–8. 10.1111/j.1438-8677.2009.00263.x [PubMed] [CrossRef] [Google Scholar]

[77] Manderson L. *How global Warming is Adding to the Health Risks of Poor People*. The Conversation. University of the Witwatersrand. Available online at: http://theconversation.com/how-global-warming-is-adding-to-the-health-risks-of-poor-people-109520 (accessed October 5, 2019).

[78] Ministers of Energy and Environment Federal/Provincial/Territorial Ministers of Energy and Environment (Canada), editor. *The Canada-Wide Acid Rain Strategy for Post-2000*. Halifax: The Ministers; (1999). 11 p. [Google Scholar]

[79] Zuhara S, Isaifan R. The impact of criteria air pollutants on soil and water: a review. (2018) 278–84. 10.30799/jespr.133.18040205 [CrossRef] [Google Scholar]

[80]https://www.aljazeera.net/encyclopedia/citiesandregions/2014/11/11/%D8%A7%D9%84%D8%AC%D8%B2%D8%A7%D8%A6%D8%B1%D8%A7%D9%84%D8%B9%D8%A7%D8%B5%D9%85%D8%A9

[81]https://sotor.com/%D9%85%D8%B9%D9%84%D9%88%D9%85%D8%A7%D8%AA-%D8%B9%D9%86-%D9%88%D9%84%D8%A7%D9%8A%D8%A9-%D8%A7%D9%84%D8%AC%D8%B2%D8%A7%D8%A6%D8%B1/

[82] https://whatis.techtarget.com/definition/Visual-Studio-NET

[83] https://www.python.org/doc/essays/blurb/

[84] https://jupyter.org/