João Gregório

FORECASTING AIRBORNE POLLUTANTS VIA TIME-SERIES MODELS

Master's Thesis in Chemistry

Chemistry Department

FCTUC

September 2017

UNIVERSIDADE DE COIMBRA

UNIVERSITY OF COIMBRA

DEPARTMENT OF CHEMISTRY

MASTER'S DEGREE IN CHEMISTRY

Forecasting airborne pollutants via time-series models

Author:

João GREGÓRIO

Advisors:

Pedro Jorge CARIDADE

Jorge Costa PEREIRA

Carla GOUVEIA-CARIDADE

2016/2017

"Every mammal on this planet instinctively develops a natural equilibrium with the surrounding environment but you humans do not. You move to an area and you multiply and multiply until every natural resource is consumed and the only way you can survive is to spread to another area. There is another organism on this planet that follows the same pattern. Do you know what it is? A virus. Human beings are a disease, a cancer of this planet."

Agent Smith, 1999 (The Matrix)

Acknowledgments

First and foremost, I would like to thank my enormous teacher advisors Pedro Caridade and Jorge Costa Pereira from the Faculty of Science and Technology of the University of Coimbra and my advisor (and supervisor) Carla Gouveia-Caridade from SpaceLayer technologies. This work was only possible with under their guidance and mentorship.

In second place I would like to thank all my coworkers at SpaceLayer technologies for taking me in and for showing me the out-of-this-world work environment that I never knew existed.

Now for the fun part... I would like to give my thanks to all my oldest (and surprisingly not yet the ugliest) friends: Filipe Cabeleira, Ana Miguel, Pedro Guilherme and especially to my closest and dear friend Ana Carrêlo. Thank you all for the companionship. I know that we don't see each other very often nowadays, but I promise you all... next time we meet, drinks are on me!

I would also like to thank my precious colleagues and friends Rui Apóstolo and Maria Guerra. May fate bring us together more often than not.

Next, my thanks goes to the three stooges and their D'artagnan: Filipe Estrada, José Miguel Sousa, José Roque and Pedro Neves. Thank you for having my back, thank you for helping out in my battles, thank you for the goofiness and thank you for letting me vent and rant without the need to apologize. We went trough a lot this past year, and even though my journey here is at its end, yours is still only half-way trough and I hope that I can be there when it ends in the same way you are here when mine does. We seriously need to go out and celebrate afterwards, and its on you! You glorious bastards!

To my better half I give the most sincere and profound "thank you" that I can concoct. Filipa Silva, thank for all that you are and that you've been to me, thank you for walking down this path by my side, thank you for helping me up when I fell down, thank you for always being there for me. Without you by my side this journey wouldn't have had half its meaning, thank you for hearing me rant, for being patient with me when I was feeling unmotivated and for never leaving my side. If

there is any greater love or dedication in this world I know not where to find them. Thank you my love!

To my family, especially to my mother and father, I can only begin to thank you all for making this possible and for enabling me to achieve all this and get this far. I know not the the words nor the sounds required to sincerely express my gratitude towards you. So I hope that you'll forgive for simple saying thank you. Thank you, from the bottom of my hearth and may it be said loud enough to reach yours. Thank you!

Agradecimentos

Em primeiro lugar, eu gostaria de agradecer aos meus enormes professores orientadores Pedro Caridade e Jorge Costa Pereira da Faculdade de Ciência e Tecnologia da Universidade de Coimbra e à minha orientadora (e supervisora) Carla Gouveia-Caridade da SpaceLayer technologies. Este trabalho só foi possivél sob a sua orientação e mentoria.

Em segundo lugar queria dar os meus agradecimentos a toda a equipa da Spacelayer technologies por me ter acolhido e por meter mostrado um ambiente de trabalho fora-deste-mundo que eu nunca sabia que existia.

Agora para a parte engraçada... Gostaria de agradecer aos meus amigos mais velhos (e surpreendentemente não os mais feios ainda): Filipe Cabeleira, Ana Miguel, Pedro Guilherme e em especial á minha amiga mais próxima e querida a Ana Carrêlo. Obrigado a todos pelo companheirismo. Sei que não nos vemos com frequência nos dias que correm mas prometo-vos... da próxima vez, as bebibas são por minha conta!

Gostaria também de agradecer aos meus estimados colegas e amigos Rui Apóstolo e Maria Guerra. Que o destino nos junte mais vezes.

O próximo agradecimento segue para os três estarolas e o seu D'artagnan: Filipe Estrada, José Miguel Sousa, José Roque e Pedro Neves. Obrigado por me protegerem as costas, obrigado por me ajudarem nas minhas batalhas, obrigado pelas palermices e obrigado por me deixarem ventilar sem ter necessidade de me desculpar. Passámos por muito neste último ano, e ainda que a minha jornada aqui esteja a chegar ao fim a vossa ainda vai a meio e eu espero poder estar presente quando ela terminar da mesma forma que vocês estão aqui presentes agora que a minha o fez. Temos seriamente de sair e celebrar depois, e são vocês que pagam! Seus bastardos gloriosos!

Para a minha melhor metade eu dou o mais sincero e profundo "obrigado" de que sou capaz de conceber. Filipa Silva, obrigado por tudo o que és e que tens sido para mim, obrigado por percorreres este caminho a meu lado, obrigado por me ajudares a erguer quando caiu, obrigado por sempre estares presente. Sem ti a meu lado esta viagem não teria tido metade do seu significado, obrigado por me ouvires

disparatar, por seres paciente comigo quando estou em baixo e por nunca deixares o meu lado. Se existe maior amor e dedicação neste mundo eu não sei onde ele se encontra. Obrigado, meu amor!

Para a minha família, especialmente para a minha mãe e para o meu pai, eu só posso começar a agradecer-vos a todos por terem feito isto possível e por me terem permitido alcançar tudo isto e chegar tão longe. Desconheço as palavras e os sons necessários para expressar de forma sincera toda a gratidão que sinto para com vocês- Por isso espero que sejam capazes de me perdoar por apenas dizer obrigado. Obrigado, do fundo do meu coração e que seja dito tão alto que alcance os vossos. Obrigado!

Abstract

Airborne pollutants pose a constant danger to human health and severely affect quality of life. Citizens breathe man-generated compounds such as carbon monoxide, nitrogen dioxide, tropospheric ozone, sulfur dioxide and other small particles that have adverse effects on their health.
Reducing the emissions of these compounds takes great effort and time, and while doing so people continue to be exposed to them. Thus mitigating their effects by reducing exposure time is a critical challenge that requires accurate forecasting models capable of predicting the slightest variations in their concentrations.
This work made use of machine learning algorithms and techniques to construct and refine linear models capable of making such predictions. In the end, there are presented models with small errors for carbon monoxide and tropospheric ozone forecasting alongside more erratic, but consistent, particulate matter and sulfur dioxide forecasting models. The one compound that the developed model was unable to forecast accurately was nitrogen dioxide. Considerations the future of this research and possible improvements are also supplied.

Resumo

Os poluentes atmosféricos representam um perigo constante para a saúde humana e afectam a qualidade de vida. Os cidadãos respiram compostos produzidos pelo homem como dióxido de carbono, dióxido de azoto, ozono troposférico, dióxido de enxofre e outras pequenas partículas que têm efeitos adversos na sua saúde.
Reduzir as emissões destes compostos requer um grande esforço e tempo, e enquanto isso as pessoas continuam expostas a eles. Assim, mitigar os seus effeitos reduzindo o tempo de exposição é um desafio crucial que requer modelos de previsão capazes de prever as variações mínimas nas suas concentrações.
Este trabalho fez uso de algoritmos e técnicas de "machine learning" para construir e refinar modelos capazes de fazer tais previsões. No fim, são apresentados modelos com erros mínimos para prever monóxido de carbono e ozono troposférico bem como modelos erráticos, mas consistentes, para realizar a previsão de matéria particular e dióxido de enxofre. O único composto para o qual o modelo desenvolvido não foi capaz de gerar previsões precisas foi o dióxido de azoto. Considerações sobre o futuro desta pesquisa bem como possíveis melhorias são também apresentadas.

List of Figures

List of Tables

Index

Preface

Air pollution is one of the biggest environmental concerns in present days affecting people all around the world, in developed and developing countries alike. The World Health Organization (WHO) estimates that in 2012 up to 3.7 million premature deaths all around the world could be blamed on air pollution [1–3]. It can be ranked in one of two different categories depending on its emission sources: background or anthropogenic pollution. The first of which occurs naturally from phenomena such as radiological decomposition, forest fires and volcanic activity whereas the latter is generated by human activity, mainly from fossil fuel burning. While anthropogenic air pollution has existed for almost as long as humankind, it has seen an exponential increase during past couple of centuries, due to the industrialization process [4, 5].

From the numerous components that build up air pollution as a whole, there are six that stand out as the most dangerous to human health according to current medical and scientific knowledge. These are carbon monoxide (CO), nitrogen dioxide (NO_2), tropospheric ozone (O_3), sulfur dioxide (SO_2), and fine particles with an aerodynamic diameter smaller than $10\,\mu$m (PM_{10}) and $2.5\,\mu$m ($PM_{2.5}$) [4, 6, 7]. Human physiology is adversely affected when exposed to any of these pollutants by either inhalation or skin contact. Each pollutant has different effects on the human body, that vary according to exposure time and severity, which will be discussed later. However, regulatory agencies are in agreement that general effects of exposure to air pollution range from irritation of the skin and mucous membranes to development of chronic diseases such as asthma. The latest "European environment - state and outlook 2015" report (SOER 2015) published little over two years ago, by the European Environment Agency (EEA) makes notice of this problem by stating that "*Air pollution is the top environmental issue for premature death*

and with impact on productivity and health. Citizens often breathe air that does not meet standards, with major sequels: asthma, chronic obstructive pulmonary disease, and cancer. Other diseases are triggered by pollution: rhinitis, conjunctivitis and dermatological disorders" [8].

Upon the realization of the dangers posed by anthropogenic air pollution, in 1979 the United Nations Economic Commission for Europe (UNECE) created the Convention on Long-Range Transboundary Air Pollution (CLRTAP) which begun to be enforced four years later in 1983 with the aim of reducing anthropogenic air pollution [9]. Positive results from the guidelines set by this convention for air pollution management and mitigation can be found in decision 2010/18 taken by the UNECE in 2010 which states that " *The Convention on Long-range Transboundary Air Pollution has delivered demonstrable improvements in reducing acidification of the environment, in reducing the highest peak levels of ozone and photochemical smog, and has begun to make improvements in atmospheric levels and deposition of nitrogen*". Despite that, the air breathed by citizens is still bellow standards and will most likely remain so for next few decades [8, 10].

Predicting the variations of atmospheric variables despite not being a permanent solution for this problem could be a great tool for mitigating the effects of anthropogenic air pollution in humans. In essence, it would enable citizens to take a proactive stance when dealing with air pollution by avoiding the more polluted areas or taking preventive medication. Tough the idea of forecasting variations in the concentration of airborne contaminants is not a novel one, most models developed and used for this purpose are deterministic and thus are extremely limited and flawed given the chaotic nature of Earth's atmosphere. This work explores the idea that locally trained machine learning data driven algorithms could be used as a non-deterministic alternative for the same purpose.

The European Space Agency (ESA) under the Copernicus - The European Earth Observation Programme [11] supplied one year worth of data collected over the city of Coimbra, Portugal, to use to train the machine learning algorithms. Making use of this data, a series of linear regression based algorithms were developed and compared with each other for the purpose of integrating a moving sensor network designed to warn people beforehand of local pollution spikes in the urban region of the city of Coimbra. The only limitation of the algorithms lie in the underlying linear character they share which is a trade-off between accuracy and model simplicity.

Introduction

2.1 Airborne pollutants

An airborne substance is classified as an air pollutant if it causes some sort of adverse effect in either the human species or the ecosystem. Its origin can be either natural or anthropogenic, the latter being the most relevant. In addition, air pollutants can also be classified as primary or secondary pollutants. The first category refers to pollutants that are directly emitted from a process, like carbon monoxide, nitrogen dioxide and sulfur dioxide. The second refers to pollutants that are not directly emitted but instead are products of reactions involving primary pollutants, such as tropospheric ozone and most particulate matter [4]. Throughout the following sections the specifics of carbon monoxide (CO), nitrogen dioxide (NO_2), tropospheric ozone (O_3), sulfur dioxide (SO_2), and fine particles (PM_{10} and $PM_{2.5}$) will be presented regarding emission, dangers to human health and regulations.

2.1.1 Carbon monoxide

Carbon monoxide is a product of the incomplete combustion of organic matter which occurs when oxygen is scarce, and it is mostly produced by the burning of carbonaceous fuels inside car engines and power plants [12]. Equation 2.1 shows an example of a complete combustion while equation 2.2 represents and incomplete combustion where it lacks enough oxygen. Not only is carbon monoxide extremely toxic to most animal species, including humans, but it is also colorless, odorless and tasteless which makes it not only very dangerous but also very hard to detect. Worldwide, carbon monoxide, is regarded as the leading cause for the most common and fatal type of air poisoning: carbon monoxide poisoning [13] .

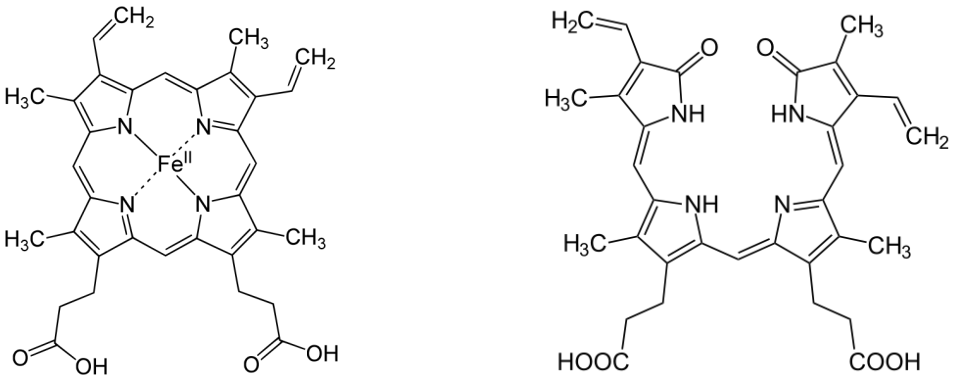


Figure 2.1: Heme b (left) and biliverdin (right).

$$CH_4 + 2\ O_2 \rightarrow\ \ CO_2 + 2\ H_2O + \text{energy} \tag{2.1}$$

$$4\ CH_4 + 7\ O_2 \rightarrow 2\ CO + 2\ CO_2 + 8\ H_2O + \text{energy} \tag{2.2}$$

Equation 2.3 shows that carbon monoxide is naturally produced in the human body by the action of heme oxygenase 1 and 2 when breaking down the heme group in hemoglobin to form biliverdin, both of which presented in figure 2.1. Despite that, the concentration in which is formed is negligible, when compared with the concentration breathed in, and it is easily ventilated under normal circumstances where ambient levels are low [14, 15].

$$\text{Heme b} + 3O_2 + \frac{3}{2}\text{NADPH} + \frac{3}{2}H^+ \rightarrow \text{biliverdin} + Fe^{2+} + CO + \frac{3}{2}NADP^+ + 3H_2O \tag{2.3}$$

However, when ambient levels are high enough it becomes harder for the human body to ventilate the amounts that enter the blood stream trough respiration. This happens because the affinity between carbon monoxide and the heme group in hemoglobin is very high due to a π-backbonding or backdonation effect. Carbon monoxide has negative formal charge on the carbon atom, however since carbon is a very electropositive atom it has a lot of stress when part of a carbon monoxide molecule that can be relieved by backbonding with the iron atom in the heme group of Hemoglobin (Hb) when forming carboxylhemoglobin (COHb). Looking at the molecular orbital diagram for the carbon monoxide molecule shown in figure 2.2,

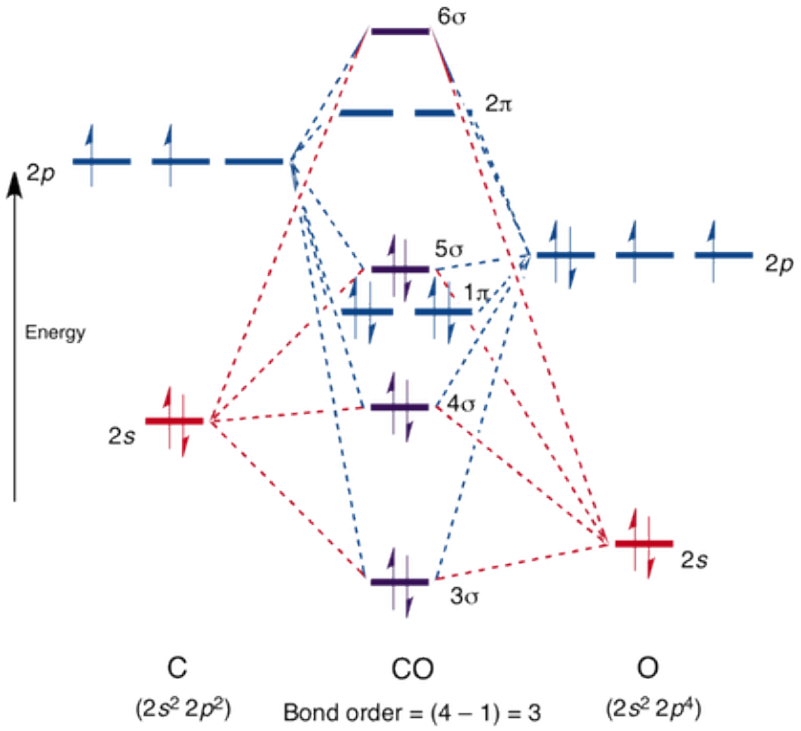


Figure 2.2: Molecular orbital diagram for the carbon monoxide molecule.

one of the non-bonding electrons in the 5σ orbital will be promoted to the 2π non-bonding orbital when near the iron atom to form the π-backbond which allows for some negative charge to move away from the carbon atom stabilizing it [16].

The fact that carbon monoxide has an affinity with hemoglobin of about 210 to 230 times greater than that of oxygen means that it binds much more strongly with hemoglobin which implies that its diffusion from the alveoli to the blood stream happens at a much faster rate than that of oxygen which has no stabilizing backbonding effect with hemoglobin and also that once binded to form carboxyhemoglobin it wont be easily removed. This has a few very important biological implications when considering that the function of Hb as an oxygen carrier is compromised when it becomes COHb. [12] Since COHb is not able to properly deliver oxygen, when at least 20 % of body Hb has been converted to COHb the oxygen delivery system in the human body becomes heavily compromised causing cells in different types of tissues throughout the body to enter a state of hypoxia and eventually suffer apoptosis. Tissues from the cardiovascular and central nervous systems are the most affected and lead to most symptoms [12, 13, 17–19].

Table 2.1: Carbon monoxide concentrations, COHb levels, and associated symptoms. Data from Goldstein et al. 2008

CO concentration (ppm)	**COHb level (%)**	**Signs and symptoms**
35	<10	Headache and dizziness within 6 to 8 h of constant exposure
100	10	Slight headache in 2 to 3 h
200	20	Slight headache within 2 to 3 h; loss of judgment
400	25	Frontal headache within 1 to 2 h
800	30	Dizziness, nausea, and convulsions within 45 min; insensible within 2
1600	40	Headache, tachycardia, dizziness, and nausea within 20 min; death in less than 2 h
3200	50	Headache, dizziness, and nausea in 5 to 10 min; death within 30 min
6400	60	Headache and dizziness in 1 to 2 min; convulsions, respiratory arrest, and death in less than 20 min
12800	70	Death in less than 3 min

Symptoms of carbon monoxide exposure, presented in table 2.1, are non-specific and highly dependent on the amount of time and concentrations of the pollutant that the organism is exposed. Prolonged exposure to low concentrations can cause persistent headaches, lightheadedness, depression, confusion, memory loss, nausea and vomiting and also increases the risk of developing and worsen cardiovascular symptoms. Beyond that, not only does acute exposure to high levels induces the same symptoms in a much shorter time span as chronic exposure, but also leads to loss of judgment, convulsions, increased heart rate, respiratory arrest, unconsciousness and death. While it is possible to recover from acute carbon monoxide poisoning, recurring to hyperbaric oxygen therapy which consists in giving patients 100 % oxygen to breathe, the momentary deprivation of oxygen to the brain often leads to the manifestation of delayed permanent and non-permanent neurological conditions such as memory loss, dementia, amnesia, psychosis and depression [17, 18, 20].

Carbon monoxide measurements often express its concentration in parts per million (ppm) and regulations imposed by the World Health Organization are set to try and keep citizens with a percentage of COHb bellow 2.5 % to avoid most, if not all, symptoms of CO exposure [4].

2.1.2 Nitrogen dioxide

Nitrogen dioxide is a naturally occurring gas coming from bacterial respiration, volcanoes and lightning. This makes it a trace gas in Earth's atmosphere where plays a role in absorbing sunlight and regulating the troposphere's chemistry, specifically serving as a catalyst for the ozone decomposition process shown by equations **??** trough 2.4 [21, 22]. However, it is also produced, in much greater amounts, by man-made activities such as fossil fuel burning. This makes motor vehicle traffic the major contributor to its emissions and presence in the lower atmosphere [4].

$$
\begin{aligned}
O_3 &\rightarrow O_2 + O \\
NO + O_3 &\rightarrow NO_2 + O_2 \\
NO_2 + O &\rightarrow NO + O_2 \\
\hline
2O_3 &\rightarrow 2O_3
\end{aligned}
\tag{2.4}
$$

Inhalation of nitrogen dioxide causes its diffuse into the epithelial lining fluid (ELF) where it metabolizes into reactive nitrogen species (RNS) and reactive oxygen species (ROS) [23]. These species cause severe damage to tissues by inducing the lipid peroxidation of lipid molecules in cellular membranes and by interfering with the biological availability of nitric oxide, the main endothelium-derived relaxing factor (EDRF) in the human body, inhibiting the proper vasodilating and vasoconstricting of the endothelium leading to endothelial dysfunction [24–27].

In a lipid peroxidation process the free radicals capture electrons from the lipids in cell membranes causing a chain reaction in which lipid peroxides (LOPs) are formed. The reaction mechanism, shown in figure 2.3, begins with the initiation step which consists in the reaction of a RNS or ROS with a reactive hydrogen from a (poly)unsaturated lipid resulting in the production of a hydrogenated molecule and a fatty acid radical. Subsequently, the unstable fatty acid radical reacts with molecular oxygen producing peroxyl-fatty acid radical, another unstable specie, that afterwards reacts with a fatty acid creating a different fatty acid radical and either a lipid peroxide or a cyclic peroxide. This cycle continues indefinitely until termination is achieved by reaction of two radical species which only happens when

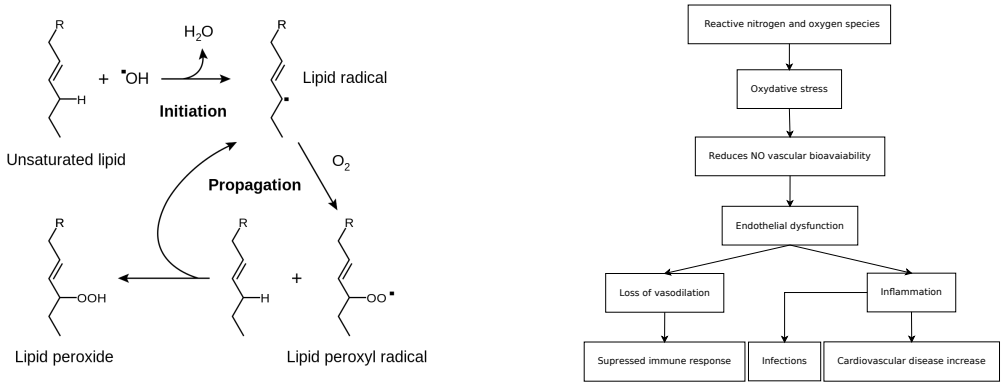


Figure 2.3: Lipid peroxidation chain reaction mechanism (left) and endothelial dysfunction biological pathway (right)

the concentration of radicals is high enough for there to be a high chance of collision between two of them. This process results in irritation of the bronchioles and alveoli and destruction of epithelial cells which causes bronchoconstriction and airway hyperresponsivness, both of which aggravate asthma symptoms, and causes the accumulation of fluids in the lungs that causes pulmonary edema [26].

Endothelial dysfunction, whose biological pathway is presented in figure 2.3, is attributed to either impaired production of NO by the endothelium or to an increased inactivation of NO by reactive nitrogen and oxygen species. The reaction of NO in the epithelium with superoxide from the decomposition of inhaled nitrogen dioxide produces peroxynitrite which is both a nitrating and oxidizing agent. Meaning that not only does this reaction depletes the vascular bioavailable nitric oxide, inducing epithelial dysfunction, but it also contributes to the lipid peroxidation process. Symptoms from this reaction include inflammation of the affected areas and loss of vasodilation which compromises immune response leading to further infections and increased risk of cardiovascular diseases [27].

Besides the negative direct implications of nitrogen dioxide in human physiology, it is also responsible for the formation of secondary pollutants such as tropospheric ozone and nitrate aerosols. Under UV light it reacts forming nitric oxide and atomic oxygen, which is extremely reactive, that in turn reacts with molecular oxygen forming ozone, whose implications will be discussed in detail in the following section. Nitrate aerosols are formed by the reaction of the nitrate ion, formed by the reaction of nitrogen dioxide with ozone, with hydrocarbons and are classified as fine particles which will also be discussed in detail further ahead [28–30].

Like carbon monoxide, nitrogen dioxide concentrations are also expressed in ppm and the World Health Organization has set a limit of exposure of a maximum of one hour in areas where NO_2 levels are equal or higher than 200 ppm and of 24 hours for levels equal or higher than 40 ppm to avoid acute symptoms. However, prolonged exposure to concentrations as low as 5 ppm can induce chronic symptoms which include the development of respiratory and cardiovascular diseases [4, 30].

2.1.3 Tropospheric ozone

Contrary to carbon monoxide and nitrogen dioxide discussed so far, ground level ozone is a secondary pollutant since it is not produced directly by human activity but instead it is formed by the reaction of compounds that are directly formed [4, 30]. It is an unstable allotrope of molecular oxygen, being constituted by three oxygen atoms bonded in an angular geometry similar to a water molecule. This means that ozone can be represented by a resonance hybrid, shown in the left side of figure 2.4, in which both structures have the center oxygen forming a double and a single bond which implies that it has a local positive charge in contrast to one of the side oxygen atoms that has a local negative charge. That said, the ozone molecule has a dipole moment and the locally positive oxygen accounts for its instability [31].

Ground level ozone should not be mistaken for stratospheric ozone responsible for building up the ozone layer [32]. Instead, ground level ozone is produced mainly by the photochemical reactions, presented in the right side of figure 2.4, of incomplete combustion products such as nitrogen dioxide and carbon monoxide in the atmosphere, earning the molecules the name of ozone precursors, and has many negative implications in human health [4, 33, 34]. While the ozone in the ozone layer absorbs UV radiation shielding us from the harmful effects of direct sunlight exposure, ground level ozone can harm lung function and irritate the respiratory system by causing or aggravating conditions such as asthma and bronchitis and even inducing hearth attacks [4, 35–37].

Being a reactive oxygen specie formed by nitrogen dioxide decomposition, it shares with it many of its negative physiological effects when inhaled [34]. Ozone reacts readily with organic double bonds inducing the lipid peroxidation of cells it enters contact with, which leads to inflammations, derived from mass cellular death, in the respiratory system with repercussions throughout the cardiovascular system.

Being a reactive oxygen specie, it also affects the vascular dilation and contraction mechanisms which causes hardening of arterial walls causing arteriosclerosis. Extreme and prolonged exposure can cause mass oxidation of tissues leading to more severe conditions such as the appearance of cancer and death [4, 38].

2.1.4 Sulfur dioxide

Sulfur dioxide has the molecular formula of SO_2 and has an angular geometry. It has a central sulfur atom double-bonded with two oxygen atoms with a slight angle. The length of the bonds is of 143.1 pm and the angle is of 119º degrees. At the standard atmosphere it is a toxic gas with a pungent and irritating smell [39].

Though volcanic activity naturally releases it into the Earth's atmosphere in trace amounts, man-made sources for sulfur dioxide emissions due so in far greater quantities. The main man-made sources for this particular pollutant are fossil fuel burning power-plants and industrial facilities. Smaller sources include industrial processes, such as metal ore extraction, sulfuric acid manufacturing and vehicle exhausts [4].

Short- and long-term exposure has adverse effects on human health. Chronic exposure is linked systemic cell apoptosis caused by sulfur dioxide and its biological derivatives while acute exposure directly affects the respiratory system causing bronchoconstriction and inflammation of the airways causing coughing, wheezing and shortness of breath which persist over time. All these acute symptoms are induced by the rapid cellular damage that takes place due to the same DNA detriment that causes the long-term symptoms [4].

In constrast with nitrogen dioxide and tropospheric ozone, sulfur dioxide is a strong reducing agent and is also very soluble in water. When inhaled, it rapidly

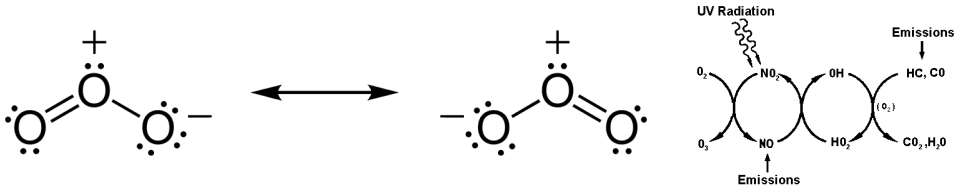


Figure 2.4: Ozone resonance lewis forms (left) and tropospheric ozone formation process (right).

reaches the alveoli in the lungs entering the blood stream where it gets hydrated [40]:

$$SO_2 \cdot nH_2O \tag{2.5}$$

And forms its biological derivatives, sulfite and disulfite [40]:

$$SO_2 + 2H_2O \rightleftharpoons HSO_3^- + H_3O^+ \tag{2.6}$$

$$HSO_3^- + H_2O \rightleftharpoons SO_3^{2-} + H_3O^+ \tag{2.7}$$

which then, stand at an equilibrium of three molecules of sulfite per molecule of disulfite in the blood stream. When inside the blood stream, both the hydrated sulfur dioxide and its derivatives are able to reach and permeate cells throughout the body and negatively affect DNA synthesis by impeding cell growth and mitosis and inducing cell apoptosis: resulting in systemic tissue damage. The exact mechanisms by which DNA synthesis is affected not not yet known, however there is evidence provided by several recent epidemiological studies that support this claim [40–44].

Given the severity of the ailments caused by sulfur dioxide exposure, the World Health Organization indicates that the maximum limit for a 24 hour exposure is of $125\,\mu g/m^3$ without even considering the dangers of other present particles [4].

2.1.5 Particulate matter

Particulate matter, made of fine particles from organic and inorganic sources with aerodynamic diameters smaller than $10\,\mu m$ (PM10) and than $2.5\,\mu m$ (PM2.5), size represented in figure 2.5 [45], are considered the most threatening airborne pollutants to human health, affecting more people than any other [1, 3, 4, 46, 47].

The formation of such particles can be divided in two main sources [48]: primary sources attributed in urban areas to road traffic, such as, carbonaceous compounds from exhaust emissions [49], re-suspension of road dust [50], tire abrasion [51] and other combustion processes; and secondary sources ascribed to the condensation of vapors or chemical reactions such as the atmospheric oxidations of SO_2 to H_2SO_4, and NO_2 to HNO_3 [52]:

$$2\,SO_2 + O_2 + 2\,H_2O \rightleftharpoons 2\,SO_3 + 2\,H_2O \rightarrow 2\,H_2SO_4 \tag{2.8}$$

$$3\,NO_2 + H_2O \rightleftharpoons 2\,HNO_3 + NO \tag{2.9}$$

Due to their small size, when they are breathed in, fine particles are able to penetrate deep into the lungs and even be absorbed into the blood stream causing damage to the organism. The level of injury they can cause varies widely depending on their concentration and type [1, 6, 53]. Given the nature of their absorption, the damage they cause is mostly focused on the respiratory system, although the cardiovascular and neurological systems can also be affected by proxy if the particles are extremely small and hazardous [54–56]. Some of the less severe short-term effects include irritation of the mucous areas like eyes, nose and throat, headaches and nausea which disappear with time. However chronic exposure to high levels is also linked to more serious conditions and can cause upper respiratory infections like bronchitis and emphysema [1, 53]. Regarding long-term effects PM exposure is also linked to chronic respiratory diseases such as asthma and lung cancer [54], cardiovascular ailments [7, 55] and brain damage [56].

Current WHO regulations regarding air quality state that PM exposure does not have a minimum limit in which no effects are noticeable. Hence, there are no strict guidelines concerning individual exposure, however there are goals for reducing the levels of ambient particles based on studies using PM2.5 and PM10 as indicators: local 24 hour levels should be kept bellow $50\,\mu g/m^3$ not being permitted to exceed it for more than seven times a year [4].

2.2 Data and programming

Data, alongside information, knowledge and wisdom, is an abstract concept that makes part of the learning process. Data is comprised of sets of values of variables about a specific subject that on their own have no particular meaning. However, they can be measured, collected and analyzed in order to be visualized creating information about that subject. Such information can be studied and used in decision-making creating knowledge on the subject. Lastly, application of this knowledge and understanding when it can be useful or not is the definition of wisdom [57].

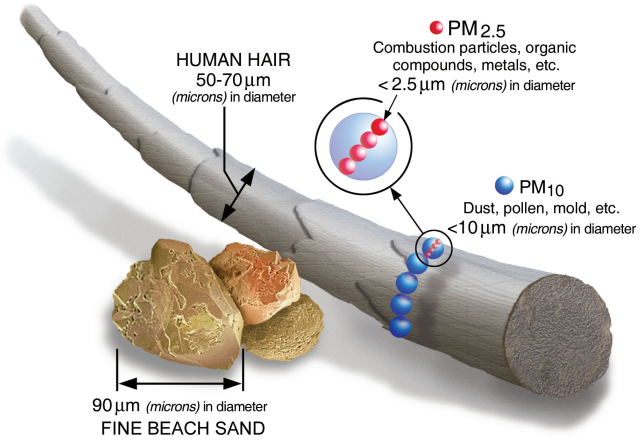


Figure 2.5: Particulate matter size comparison.

The first time the word "data" was used to define "transmittable and storable computer information" was in 1946 followed the term "data processing" in 1954 to define "the collection and manipulation of items of data to produce meaningful information" before the dawn of the *Digital Revolution* which marked the beginning of the *Information Age*. Before then, our ability to store data was confined to using *analog storage* (paper, film, audiotape, vinyl and VHS tapes), however it has grown exponentially since then due to the creation of *digital storage* means (servers, mainframes, hard-drives, mini-disks, CD's and DVD's among others) as can be seen in figure 2.6 [57].

This sudden and exponential increase in storage capacity enabled the rise of bigger and more complex data sets that are far beyond the reach of traditional data processing methods which are unable to retrieve any useful information from them within a tolerable elapsed time. The term used to describe such data sets is *Big Data* [58].

Big data is characterized based on three key aspects; being volume, variety and velocity [59]. The first of which refers to the quantity of data that it is being generated and stored, the second to its type and nature, and the last to speed at which it is generated and processed. The sheer volume of big data is simultaneously its biggest drawback and advantage. The bigger the data set, the more information can be derived from it and the more insightful it can be, however it also makes it difficult to manage and analyze properly. Variety and velocity are also intrinsic

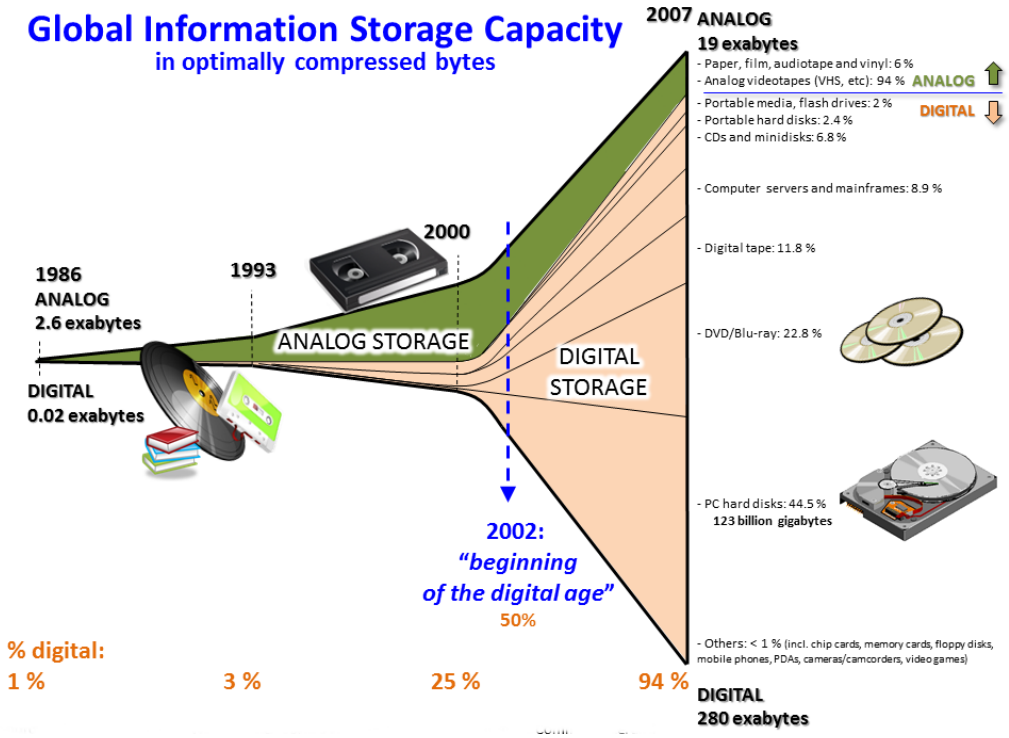


Figure 2.6: Info-graphic showing the growth of global storage capacity with the rise of digital storage.

characteristics of big data to differentiate it from *pre-digital data* that was mostly comprised of text and log files generated at a slow and steady pace. Nowadays, data comes in many different formats, ranging from videos to simulations and is generated at a much faster pace due to the rise of smart technologies and sensor networks [57].

The connection between big data and machine learning is an obvious one. The later is designed to handle large volumes of data to discover patterns and infer functions to retrieve information from it without an explicit need to understand the former. The inner workings of machine learning techniques are going to be discussed in the following section, however they can derive sense from large amounts of data, independent of its typing, in near-real time making data mining and predictive analytics easier than with traditional data-managing techniques [60].

2.2.1 Time series

Time series are a special type of data set, that is characterized by having its data points indexed in time order: a sequence of discrete-time data. Similarly to any data set, depending on its volume it can be considered a big data set. It differs from other data sets because of this natural temporal ordering, which means that the same analysis techniques employed for cross-sectional studies, where it does not exist any natural ordering, cannot be employed [61]. Traditionally the goal of analyzing time series in the context if meteorology is for forecasting purposes and it is comprised of two stages: stationarity verification and curve fitting.

A stationary process, by definition is one where its probability distribution does not change when shifted in time, meaning that parameters such as variance and mean remain constant. Most statistical procedures used during analysis assume the stationarity of the times series, so it is necessary to verify that it is in fact stationary, by using the Dickey-Fuller test [62], before the application of any statistical method and if it is shown that it is not the series needs to be transformed in order to become stationary [63].

Curve fitting consists in constructing a mathematical function that best fits the data series while subjected to constraints. It can be either interpolation, when an exact fit for the function is required, or smoothing, when only an approximate function is needed. For the purposes of time series forecasting the later is usually more commonly used and can be done by measuring the rolling statistics or by applying a filter such as the Savitzky–Golay one [64].

Dickey-Fuller test A time series can be seen as an autoregressive (AR) model since it is a representation of an apparent random process in which the output variables depend linearly of its own previous values. As such, the Dickey-Fuller test can be employed in order to verify if the null hypothesis, indicated by the presence of a unit root, is present in the model [62].

A unit root, ρ, is a feature whose presence in some stochastic processes causes problems when inferring statistical properties of a time series model. To check for its presence in a time series, this test considers an AR(1) process in which only the

previous term, y_{t-1}, and the noise term, u_t, contribute for the output, y_t [63, 65]:

$$y_t = \rho y_{t-1} + u_t \tag{2.10}$$

And states that if $\rho = 1$ a unit root is present since that would make the model non-stationary because the output term would be equal to the previous term plus the noise. However, using this equation directly poses a problem: under the null hypothesis both the previous and the output term as non-stationary. To address this constraint, the last equation can be rewritten as:

$$\begin{aligned} y_t - y_{t-1} &= (\rho - 1) y_{t-1} + u_t \\ \Delta y_t &= \delta y_{t-1} + u_t \end{aligned} \tag{2.11}$$

Meaning that, under the null hypothesis, the y_{t-1} on the right-handed side of the equation would disappear while leaving the same term on the left-handed side unchanged and stationary.

For the actual testing, the Student's t-test cannot be used, under the null hypothesis being true, because y_{t-1} is itself non-stationary so the ordinary central limit theorem (CLT)[1] does not hold with the least squares estimators for δ [66]. However, the asymptotic distribution for the least squares estimators for δ under the null hypothesis have been calculated and tabulated by David Dickey and Wayne Fuller in 1979 (hence the name of the test) so they can be compared with the values of the t-statistic [62]: if the test statistic value is less than the relevant critical value then the null hypothesis is rejected meaning that the series is stationary and if it is not then the null hypothesis cannot be rejected and the series is non-stationary which means that it requires further pretreatment before it an be properly analyzed [62].

Moving average A moving average is calculated by dividing the data into fixed-sized sequential subsets and measuring and plotting its averages, making it a reactive quantity since it reacts to the data that is already established [61]. The shorter the size of the subsets the more sensitive is the signal on contrast with larger subset

[1]The central limit theorem states that in common scenarios, when independent random variables, with well-defined expected values and variances, are added their sums will tend towards a normal distribution regardless of the underlying distribution of the original samples.

sizes. Assuming a window size of value n, the average, $\bar{\mathrm{Y}}_{SM}$, for a given subset is given by [67]:

$$\bar{\mathrm{Y}}_{\mathrm{SM}} = \frac{\mathrm{Y}_m + \mathrm{Y}_{M-1} + \cdots + \mathrm{Y}_{M-(M-1)}}{n} = \\ = \frac{1}{n}\sum_{\mathrm{i}=0}^{n-1} \mathrm{Y}_{M-\mathrm{i}} \tag{2.12}$$

And for calculating new successive values, a new value, $\frac{\mathrm{Y}_M}{n}$, is added into the sum and the last of the old values, $\frac{\mathrm{Y}_{M-n}}{n}$, is excluded:

$$\bar{\mathrm{Y}}_{SM} = \bar{\mathrm{Y}}_{SM,\mathrm{prev}} + \frac{\mathrm{Y}_M}{n} - \frac{\mathrm{Y}_{M-n}}{n} \tag{2.13}$$

This comprises the basis for a moving average, the simplest of all moving average filters. It can be further improved upon by adding a weight factor according to the distance of each data point. This can be done by applying a weighting factor that exponentially decreases for older data: exponential moving average (or exponentially weighted moving average). To create such a filter the weights are defined as [68]:

$$\{w_1, w_2, \cdots, w_k\} \rightarrow \sum_{n=1}^{k} w_n = 1 \tag{2.14}$$

And then they are incorporated into the moving average formula, transforming equation 2.12:

$$\bar{\mathrm{Y}}_{SM} = \frac{1}{n}\sum_{n=1}^{k} w_n \mathrm{Y}_{M-\mathrm{i}} \tag{2.15}$$

In practice the weights are chosen to give more relevance to more recent data without completely discarding older data points.

Moving average techniques smooth out short-term fluctuations and highlight longer-term trends or cycles. They are only limited by the fact that they require at least a certain number of observations being made, corresponding to the window size, before being employed. Moreover they are a very easy and useful method for smoothing time series prior to forecasting being capable of improving accuracy [61].

Savitzky–Golay filter It is designed to be applied on a set of data points with the intent of smoothing the data by increasing the signal-to-noise ratio[2] without distorting the original data signal. The smoothing is done by a process of convolution[3] in which successive sub-sets of sequential data points are fitted with a low-degree polynomial using the least squares method.

For equally spaced data points, such as those present in most time series, the least squares approach yields an analytical solution in the form of a set of coefficients, aptly named *convolution coefficients*. Considering a data set of n equally spaced data points (x_i, y_i) a polynomial is fitted, via least-squares to a subset of m (odd-numbered) sequential data points separated by an interval h [67]:

$$z = \frac{x - \bar{x}}{h} \tag{2.17}$$

in which a change of variable is performed, taking $\bar{x}$ as the central point. By definition z can take the values of:

$$z \to \left\{ \frac{1-m}{2}, \cdots, 0, \cdots, \frac{m-1}{2} \right\} \tag{2.18}$$

And the polynomial of degree k can be defined as:

$$\mathrm{P} = \mathrm{c}_0 + \mathrm{c}_1 z + \mathrm{c}_2 z^2 + \cdots + \mathrm{c}_\mathrm{k} z^k \tag{2.19}$$

Which can be used to construct and solve normal equations yielding the values of the convolution coefficients:

[2]Abbreviated by SNR, it is a measure used for comparing the level of a signal with the level of its background noise. Any ratio greater than 1:1 indicates that there is more signal information contained in the data than noise.

[3]Integral transform using the product of two functions after one of them is reversed and shifted [67]:

$$\begin{aligned}(f \star g)(t) &= \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \\ &= \int_{-\infty}^{\infty} f(t-\tau) g(\tau) d\tau \end{aligned} \tag{2.16}$$

$$\mathbf{c} = (\mathbf{J^T J})^{-1}\mathbf{J^T y}$$
$$\mathbf{C} = (\mathbf{J^T J})^{-1}\mathbf{J^T} \tag{2.20}$$

So they can take the general form of:

$$\mathrm{P_i} = \sum_{j=-\frac{m-1}{2}}^{\frac{m-1}{2}} \mathrm{C_j} y_{i+j} \; ; \; \frac{m-1}{2} \leq i \leq n - \frac{m-1}{2} \tag{2.21}$$

Used for tabulating every convolution coefficient based on the degree of the polynomial and the number of data points to be fitted. These are generalized and can be utilized for every situation in which the application of the Savitzky–Golay filter is required [67].

2.2.2 Python

Not a full and detailed explanation about the Python programming language, but rather a note of context to the reader.

All programming and coding performed in this project, including the construction of machine learning and visualization algorithms, was done using Python. It is a general-purpose-high-level programming language and one of the most widely used worldwide in current days. It was chosen because it is simple, it runs fast and it is possible to incorporate it as modules in programs written in other, more common languages such as C. Also, there are a number of useful tools already designed to be used with python. Some of these are designed to handle large volumes of data such as the *NumPy* package and the *Pandas data analysis library.* Others are more specific for the projects purpose such as the *Scikit-learn* toolkit which was used intensively.

2.3 Machine learning

Machine learning was defined in 1959 by Arthur Samuel, an american pioneer in the field of artificial intelligence, as the "field of study that gives computers the ability to learn without being explicitly programmed" [69]. It is a multidisciplinary

field of study and research that encompasses the principles of statistics, engineering and computer science to explore the development of algorithms capable of learning from and making predictions on data. It is particularly useful to address complex problems which cannot be currently modeled deterministically, either by lack of computational power or of knowledge about the problem, by making use of statistical notions to derive an approximate solution. Unlike traditional algorithms, that are confined to follow hard rules from static program instructions, those based on machine learning, called *learners*, are able to learn from examples and experience to make data-driven predictions, effectively turning data into information [70].

2.3.1 Terminology

Machine learning algorithms (MLA's) are often created to address a task that normally would be done by a human being [70]. A simple example would be related with weather forecasting. By looking to the sky a person is able to guess if it is going to rain or not depending on the temperature, humidity and cloud formations among other factors. Nowadays, this task is mostly done by learners, that were trained to analyze those same variables and patterns. The systems based on these learners are called *expert systems* because they replace an expert: in this case a weatherman was replaced by a computer.

Following up with the same example, the factors referred alongside the observation of the forecast (if it did rain or not) are called *features* and usually encompass everything that is known about the system. A corresponding set of features earns the name of *instance*. Learners use *training examples*, which are comprised of several instances, to find patterns and relationships between different features, including the *target variable* (which is the presence or absence of rain in this example) and use these relationships to find the target variable in an unknown instance. If the target variable is an object the learner is a *classification* algorithm and if it is a number then the learner is a *regression* algorithm.

After training the learner needs to be validated, which is done by using it on a *test set*. This set is similar to the training, however it was not used in the training phase, hence it is unknown to the learner but the users have knowledge of all features in every instance including the target variable. The learner will use the features to determine the target variable which is then compared to the real one to account

Table 2.2: Schematic representation of features and instances.

	Features				
Instances	A1	A2	A3	A4	Target variable A
	B1	B2	B3	B4	Target variable B
	C1	C2	C3	C4	Target variable C
	D1	D2	D3	D4	Target variable D

for accuracy and model performance. If the results are accurate enough then the learner is ready for real world use, if they are not then the learner requires tweaking which may be as simple as supplying it with more training data or as complicated as finding out if all used features are actually relevant or if there is a lack of known features.

This example is the human equivalent to concept learning, it shows a case of *supervised leaning* where the target variables are known and included in the training set. However, there is also the case were the target variables are unknowns, in which the correct approach to create a learner would be to use *unsupervised learning* that infers a function to describe hidden structures in data. Both these concepts are discussed in detail in the following section [70, 71].

2.3.2 Supervised learning methods

The overall concept of supervised learning, shown in figure 2.7, introduces upon the learner the task of analyzing labeled training data, which includes an input and an output part, the features and the target variable respectively, and using that knowledge to map new examples [72–74]. When the learner is confronted with the training data it tries to model it and find a function that loosely describes it by means of function approximation [75]. That function is then generalized and applied inductively to try and model similar systems that are previously unknown to the learner [70, 73]. Supervised learning consists in building models that can be taught from specific examples and generalize the knowledge they gain to model new data.

Figure 2.8 shows the hierarchy of supervised machine learning algorithms [73]. They can be divided into two different categories according to which type of response they are designed to look for. Those that produce a nominal response are called *classifiers* while those that generate a numerical value as a response are called *regression*

models [70]. Classifiers, as the name suggests, undergo the task of classifying new instances into a set of categories according to the rules or functions derived from the training set while regression models attempt to use approximate functions developed in the training phase to forecast and relate unknown (target) variables belonging to new instances. Examples of classification and regression problems respectively are assigning a given email as either *spam* or *not-spam* and temperature forecasting for a given location. While the two share the same principles, the algorithms used for each case are fundamentally different not only regarding the response they give, opposing labels to numbers [70, 73].

Classifiers can be further divided onto two sub-categories [76]. Probabilistic classifiers are highly based on statistical principles and are capable of assigning a probability distribution instead of a single label for a given instance which allows for the classification process to be done with a degree of certainty [77]. On the other hand structural classifiers take more into account the data structure to assign labels to new instances [78]. The most common probabilist classifier in use is the Naïve Bayes classifier and concerning structural classifiers the most widespread are decision trees (also know as classification trees), the k-nearest neighbors (k-NN) algorithm and support vector machines (SVM) [73].

Regression models work as an iterative process that models the relationship between variables and uses some error measuring criteria to improve after each iteration until it reaches an optimal state [70, 73, 79]. Most of the more commonly used regression algorithms are based in some form of linear [80], logistic [81] or local regression [70] and can either be univariate or multivariate in nature. A sub-class of regression algorithms includes regularization algorithms which penalize models based on their complexity, favoring simpler models that are better at generaliz-

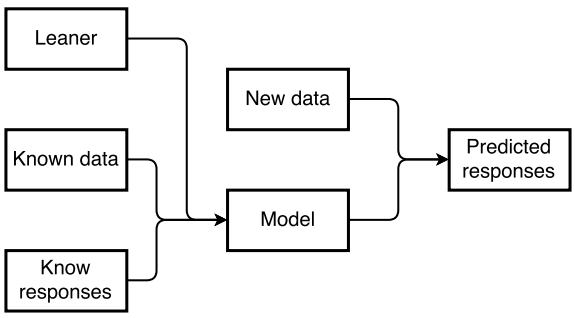


Figure 2.7: Concept of supervised learning.

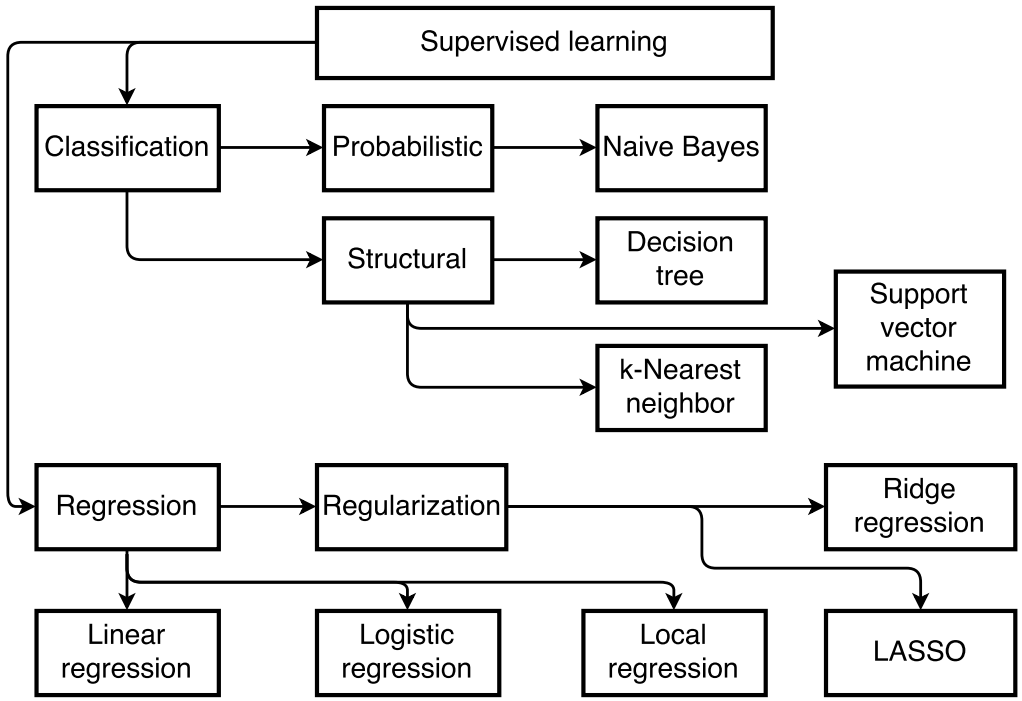


Figure 2.8: Most commonly used supervised learning algorithms.

ing [70, 82]. Though these models are regression based in essence, they are often considered separate from the bulk of regression algorithms given their usefulness. Examples are the ridge regression (RR) model and the least absolute shrinkage and selection operator (LASSO) model [70].

Naïve Bayes classifier: The fundamental assumption in this type of classifiers is that the value of any feature is independent from that of any other features, including the label. Meaning that each feature contributes in the same way for the target label value independently of the contributions of any other feature. This is a very a naive assumption, given that most real situations there exists at least some degrees of dependency between features, however, despite that the algorithm still works exceptionally well as an efficient first approach.

In order to calculate probability the algorithm uses the Bayes' theorem, written as [83]:

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}} \tag{2.22}$$

Where P(A) and P(B) are the probabilities of observing A and B without viewing the other, P(B|A) is the probability of observing A given that B is true and P(A|B) is the conditional probability of observing A given that B is true. If B is a vector of features, $B = x_1, ..., x_n$ then, by using the chain rule, it can be written as [84]:

$$\begin{aligned} P(A|x_1, ..., x_n) &\propto P(A, x_1, ..., x_n) = \\ &= P(x_1, ..., x_n, A) = \\ &= P(x_1|x_2, ..., x_n, A) \cdot ... \cdot P(x_{n-1}|x_n, A) \cdot P(x_n|A)p(A) = \\ &= p(A) \sum_{i=1}^{n} P(x_i|A) \Rightarrow \\ P(A|x_1, ..., x_n) &= \frac{1}{P(B)} p(A) \sum_{i=1}^{n} P(x_i|A) \end{aligned} \tag{2.23}$$

Where the final expression can be used on new instances to calculate a probability that, by comparison, can be converted into a label [70].

Decision trees: The original data set is broken down into smaller subsets while an associated decision tree is incrementally developed. The features and labels are arranged in hierarchical diagram, resembling a tree, where the leaves represent the class labels and the branches (connections) show the relationships between features that eventually lead up to a class label. An example of a decision tree is shown in figure 2.9, where it is used for determining point group symmetry of a given molecule [70, 85].

Support vector machines: Given a set of training examples, where each instance belongs into one of two separate categories, new examples are mapped into one of these two existent categories [87]. This is done by constructing a hyperplane that separates the training examples in space. A better separation implies a better classification for new examples and a good separation is achieved when the distance from the hyperplane to the nearest points of each category is at its maximum. This process is not limited to a 2D space, since it can be used on a high- or infinite-dimensional space, the only difference resides in the fact that more hyperplanes

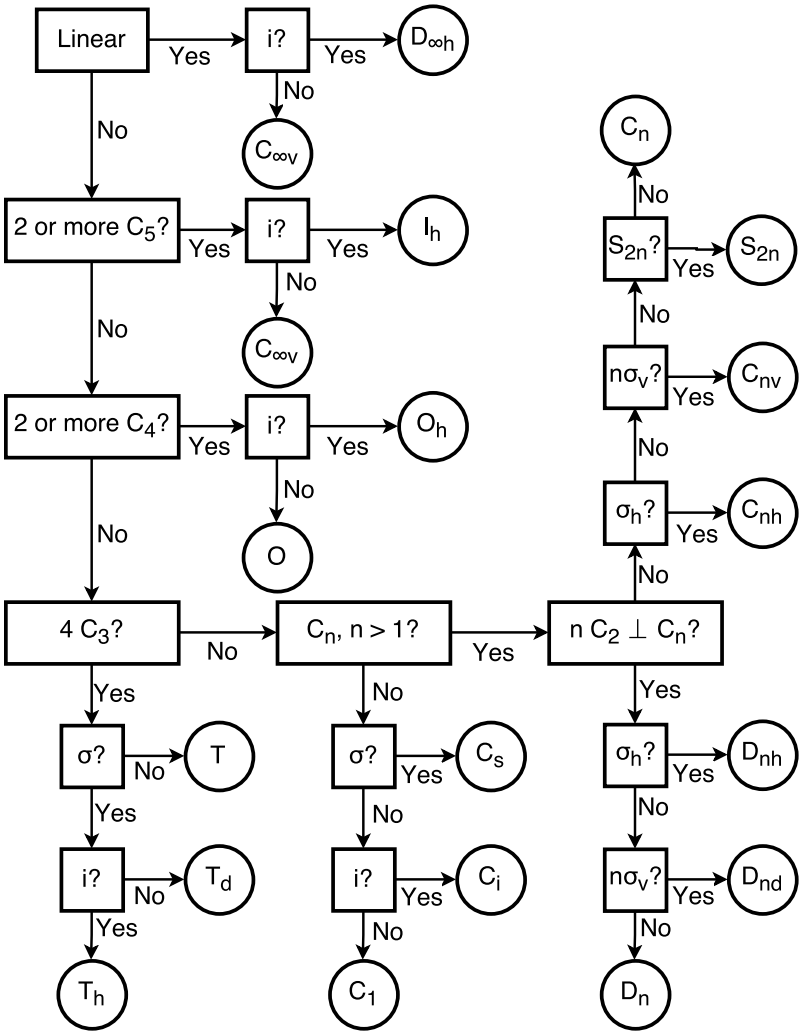


Figure 2.9: Example of a decision tree for determining point group symmetry.

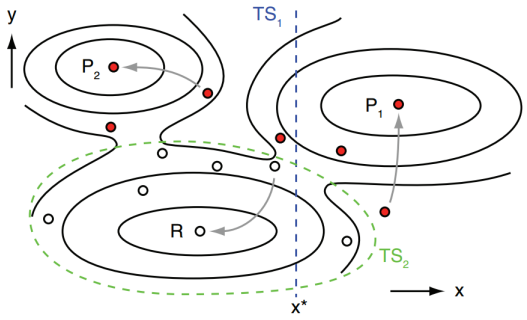


Figure 2.10: "Transition state TS1, placed along an assumed reaction coordinate x, separates reactant R and product P1 but fails to describe the transition to P2. TS2 is a surface which can be determined by training a machine to distinguish a set of points as reactant or product."Image taken from reference [86].

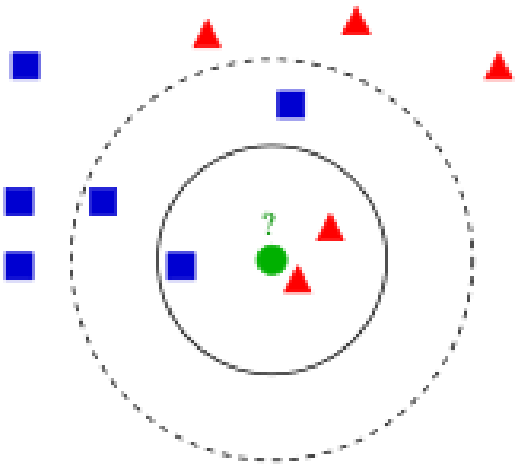


Figure 2.11: Example of kNN classification. For k=3 the assigned class would be a triangle and for k=5 the assigned class would be a square. Image by By Antti Ajanki AnAj (Own work) [GFDL (http://www.gnu.org/copyleft/fdl.html), CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/) or CC BY-SA 2.5-2.0-1.0 (https://creativecommons.org/licenses/by-sa/2.5-2.0-1.0)], via Wikimedia Commons

are necessary for separating the data points on higher dimensional spaces [88]. In chemistry this method has been successfully used to optimize transition state theory (TST) separation planes, as exemplified in figure 2.10 [86].

k-Nearest neighbors: It is used primarily for pattern recognition in cases where the features are spatially positioned. The first step is to create a feature space, that can have varied dimensions and assigning classes according to the proximity of the feature data points [89]. To evaluate a new instance, vectors are plotted from the new data point to the k-nearest neighbor points disregarding their class. The new data point takes the class belonging the the majority of its neighbors [90, 91]. That said, the parameter k can be called an hyper-parameter since it must be chosen prior to model training. The impact of the choice of k can be seen in figure 2.11. It cannot be a even number nor it can be a multiple of the number of existent classes in order to prevent ties when classifying new instances. Also, the larger it is the lesser the noise component will interfere with the classification, however it also makes the boundaries between classes less distinct [92]. In the chemistry and pharmaceutical industries kNN is used as a clustering method to find compounds of identical structures and similar properties which speeds up the first stages of the drug discovery project [93].

Linear regression: Is based on modeling the relationship between a scalar dependent variable, called y and one or more independent variables, called $X_n = \{x_1, x_2, ..., x_n\}$. When only a single independent variable is used, $n = 1$, the model is called a *simple linear regression* and, on the other hand, when more than a single independent variable is applied, $n > 1$, the model is called a *multiple linear regression* [81]. There is also the case when the dependent variable is not a scalar but instead it is a vector which implies that the dependent variable is a matrix: $X_{n,m}$. A linear regression based on this principle is called of *multivariate linear regression* (or *general linear model*) [94].

The approach to build a linear regression model passes by gathering enough data to create an overdetermined system comprised by a set of equations that largely outnumbers the unknowns. Afterwards, an iterative process based on a least squares approach takes place to find the optimal values for the unknowns that can be used for creating the predictor [95].

Mathematically there are subtle differences regarding the least squares approach for each type of linear regression:

- **Simple linear regression:** The simplest case of linear regression model. The dependent and independent variables are placed in a two-dimensional space, usually a Cartesian coordinate system, where they take the form of y and x respectively and a linear function is fitted to the sample points [80]. That function is a non-vertical straight line resembling [96]:

$$y_i = \alpha + \beta x_i \tag{2.24}$$

  where α is the y-intercept and and β is the slope. This function can predict the dependent variables as a function of the independent variables. By adding and error term, ε_i, to the model function it can be further refined to capture the deviation of the data from the model:

$$y_i = \alpha + \beta x_i + \varepsilon_i \tag{2.25}$$

  The model is trained by finding values for the parameters α and β that provide an optimal fit for the training data. This is done by means of a least-square approach [97, 98]:

$$\sum \varepsilon_i^2 = \sum (y_i - y_i^*)^2 = \\ = \sum [y_i - (\alpha^* + \beta^* x_i)]^2 \tag{2.26}$$

The asterisk marked variables denote the predicted values of x and y by the model. The values of the parameters are found by solving the minimization problem of the squared sum of residuals in the form of $\sum \varepsilon_i^2$ which yield the optimal values of [99]:

$$\beta^* = \frac{\mathrm{Cov[x, y]}}{S_x^2} \\ \alpha^* = \bar{y} - \beta^* \bar{x} \tag{2.27}$$

The optimal value of β is given by the fraction of the covariance between the dependent and independent variables and the square of the variance of the explanatory variables. And α is given by the mean of y minus the product of the mean of x by the optimal value of β calculated beforehand. These values can be substituted in the original line function giving [100]:

$$y_i = \bar{y} - \frac{\mathrm{Cov}[x, y]}{S_x^2}\bar{x} + \frac{\mathrm{Cov}[x, y]}{S_x^2} x_i \tag{2.28}$$

That can be simplified:

$$\begin{aligned} y_i &= \bar{y} - \frac{\mathrm{Cov}[x, y]}{S_x^2}\bar{x} + \frac{\mathrm{Cov}[x, y]}{S_x^2} x_i \Leftrightarrow \\ y_i &= \bar{y} - r_{xy}\frac{S_y}{S_x}\bar{x} + r_{xy}\frac{S_y}{S_x} x_i \qquad \Leftrightarrow \\ \bar{y} - y_i &= r_{xy}\frac{S_y}{S_x}(\bar{x} - x_i) \qquad \Leftrightarrow \\ \frac{(y_i - \bar{y})}{S_y} &= r_{xy}\frac{(x_i - \bar{x})}{S_x} \end{aligned} \tag{2.29}$$

where r_{xy} is the sample correlation coefficient between dependent and indepen-

dent variables and S_x and S_y are the uncorrected sample standard deviations of x and y respectively. This also shows that r_{xy} is the slope of the regression line for the standardized data points. It can also be squared in order to generate the value of the coefficient of determination, R^2 [101]:

$$\begin{aligned} R^2 = r_{xy}^2 = \\ = \left(\frac{\bar{xy} - \bar{x}\bar{y}}{\sqrt{(\bar{x^2} - \bar{x}^2)(\bar{y^2} - \bar{y}^2)}} \right)^2 \end{aligned} \tag{2.30}$$

Giving a measure about the variance magnitude existent in the dependent variable that is forecasted from the independent.

- **Multiple linear regression:** Consists on a complication of the simple linear regression model where more than one predictor, independent variable, exists. Nearly all real-world problems involve more than a single predictor. The regression equation can be written as [80]:

$$\begin{aligned} y_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} = \\ = \mathrm{X}_i^T \boldsymbol{\beta} \qquad ; i = 1, 2, \cdots, p \end{aligned} \tag{2.31}$$

where $\mathbf{X}_\mathrm{i}^\mathrm{T}$ is the transpose of the vector containing all predictors and $\boldsymbol{\beta}$ is the vector containing all the coefficients (including the α). The predictors vector is transposed in order for the final term to yield the inner product of the two vectors. Similarly to what has been done for the simple linear regression an error term can also be added to account for data deviation from the model:

$$y_i = \mathbf{X}_i^T \boldsymbol{\beta} + \varepsilon_i \tag{2.32}$$

From this equation the least squares method can also be used to minimize the error term, corresponding to the squared sum of residuals, in order to find the parameters necessary to construct a good predictor equation. Assuming that exist several dependent variables such as: $\mathbf{Y} = \{y_1, y_2, ..., y_n\}$ the vector

equation can be written as:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{2.33}$$

and in this equation all values are vectors except $\boldsymbol{X}$ which is a matrix. And since the least squares approach works by minimizing the squared sum of residuals, the quantity that needs to be minimized is:

$$\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = \sum_{i=1}^{n} \varepsilon_i^2 \tag{2.34}$$

By definition it is known that the value of ε from equation 2.33, so it is possible to expand equation 2.34 into:

$$\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (y - \mathbf{X}\boldsymbol{\beta})'(y - \mathbf{X}\boldsymbol{\beta}) \tag{2.35}$$

Which means that the minimization problem can be written as:

$$\begin{aligned} \min_{\beta}\varepsilon'\varepsilon &= (y - \mathbf{X}\boldsymbol{\beta})'(y - \mathbf{X}\boldsymbol{\beta}) = \\ &= (y' - \mathbf{X}'\boldsymbol{\beta}')(y - \mathbf{X}\boldsymbol{\beta}) = \\ &= y'y - \boldsymbol{\beta}'\mathbf{X}'y - y'\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \\ &= y'y - 2\boldsymbol{\beta}'\mathbf{X}'y + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \end{aligned} \tag{2.36}$$

Note, that in this deduction $\boldsymbol{\beta}'\boldsymbol{X}'y = y'\boldsymbol{X}\boldsymbol{\beta}$ since both terms are scalars, the transposition of the term is equal to itself. The final expression is differentiated respecting to $\boldsymbol{\beta}$ and set its derivative equal to zero:

$$\begin{aligned} \frac{\partial(\varepsilon'\varepsilon)}{\partial_{\underline{,}}\beta} &= \frac{\partial(y'y - 2\boldsymbol{\beta}'\mathbf{X}'y + \boldsymbol{\beta}'\mathbf{X}'\boldsymbol{X}\boldsymbol{\beta})}{\partial_{\underline{,}}\beta} = \\ &= -2\mathbf{X}'y + 2\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = 0 \\ \mathbf{X}'\mathbf{X}\boldsymbol{\beta} &= \mathbf{X}'y \end{aligned} \tag{2.37}$$

Afterwards, this expression can be multiplied by $(\mathrm{X}'\mathrm{X})^{-1}$ to give the least

squares estimator for the parameters [97, 98]:

$$\boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'y \tag{2.38}$$

Similarly to the simple linear regression model, after finding the parameters the function can be used as an estimator for predicting new dependent variables using known independent variables.

- **General linear model:** Contrary to the simple and multiple linear regression models discussed so far, general linear models make use of multivariate measurements to construct models able to estimate several quantities at once from a single set of estimators. In simpler terms, instead of considering a single dependent variable, y, they take into consideration several in the form a vector, $\mathbf{Y} = \{y_1, y_2, ..., y_i\}$ which makes them multivariate linear regression models.

  The general equation for a multivariate regression model can be written as [102]:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{U} \tag{2.39}$$

  where $\mathbf{Y}$ is a matrix comprised of a series of multivariate measurements, $\mathbf{X}$ is the model matrix built of the dependent variables, $\mathbf{B}$ is the coefficient matrix which contains the parameters and $\mathbf{U}$ is the noise or error matrix. In the case of $\mathbf{Y}$, $\mathbf{B}$ and $\mathbf{U}$ being vectors, the above equation relates to that of a multiple linear regression model discussed prior to this one.

  The overall deduction for the least squares approach remains the same as the one for multiple linear regression and the coefficients can be found by the equation [97, 98]::

$$\boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \tag{2.40}$$

  After finding the coefficient values for the $\boldsymbol{\beta}$ matrix the prediction process is carried out in a analogous way to that of both linear regression models discussed prior to this one.

Logistic regression: When the dependent variable, y, is categorical instead of numerical, a logistic regression models can be used as a tool for predicting such a

variable. [81] Logistic regression can be easily applied when the dependent variable is binary: can only take up to two different values, "0" or "1", that refer to outcomes of a true/false or win/lose situation. And although it is possible to model more complex systems using this regression, a more common approach resides in simplifying a particular problem until is has a binary solution [103, 104].

The basis for this regression type is a sigmoid (logistic) function. It resembles and S-shaped curve that can take any real-valued number and map its output as a value between 0 and 1 but never equal to any. It can be written as [105]:

$$y = \frac{1}{1 + \mathrm{e}^{-(\beta \mathrm{x} + \alpha)}} \tag{2.41}$$

where e is the base of the natural logarithm, Euler's number. Similar to the simple linear regression, the values of α and β define the transformation from the independent variable, x, to the dependent variable, y. These values are found by fitting the sigmoid function to the training data using a least squares approach similarly to a standard linear regression.

The y values outputted by a logistic regression model are not the class values but instead they represent the probability distribution that gives the probability of a given input belonging into the default class ($y = 1$):

$$\mathrm{P}(x) = \mathrm{P}(y = 1|x) \tag{2.42}$$

However, to obtain a strict classification from these values it is necessary to transform the probabilities into the class values of 0 or 1. This is done by splitting the sigmoid function in half regarding the vertical axis and considering all dependent values above the axis as belonging to one class and those bellow the axis a belonging to the other, such as:

$$0 \text{ if } \mathrm{P(x)} < 0.5 \tag{2.43}$$

$$1 \text{ if } \mathrm{P(x)} \geq 0.5 \tag{2.44}$$

Although the probabilities can be used directly, in order to get a straight answer for a classification problem this last step is necessary.

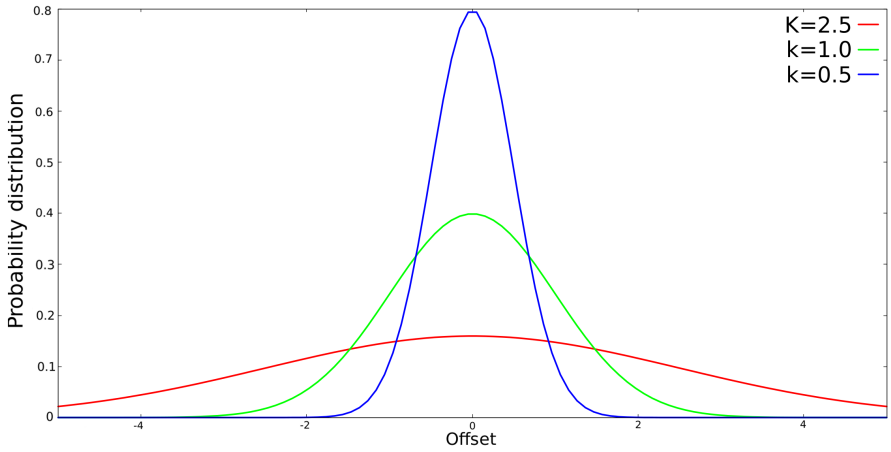


Figure 2.12: Effect of the hyper-parameter k on the normal probability distribution of a Gaussian function.

Local regression: Also called of locally weighted linear regression model [70], this class of regression models combines the simplicity and effectiveness of a simple (or multiple) linear regression model with the more complex notions of a k-nearest-neighbor styled algorithm [106, 107].

Linear regression models tend to underfit the non-linear data and even when properly trained their intrinsic training error is substantially large. And non-linear regression models (not discussed on this dissertation) are computationally very demanding and hard to implement despite yielding better results than their linear counter-parts when modeling non-linear data [70].

Local regression models locally fit linear sub-sets of a larger non-linear dataset based on a proximity criteria which is usually defined by a Gaussian function [108]:

$$w(x, x_i) = \exp\left(\frac{|x_i - x|^2}{-2k^2}\right) \tag{2.45}$$

where k is a hyper-parameter that determines how wide the base of the Gaussian function is, shown in figure 2.12. For each local fit the data points nearest to the central data point will weight more for the fit, and these values are defined by the choice of k. This allows the model to have the flexibility of a non-linear approach while still retaining the simplicity of a linear model [70].

2.3.2.1 Regularization algorithms

To better understand the role and importance of regularization algorithms it is necessary to address the *bias-variance dilemma* [109]. Both bias and variance are sources of error: error due to bias is the difference between the expected prediction of the model and the correct value which it is trying to predict while error due to variance relates to the variability of a prediction for a given data point [110]. In essence they are the opposite concepts of accuracy and precision respectively.

The mathematical definitions for bias and variance can be found trough the decomposition of the error term [77]:

$$\begin{aligned}
\mathrm{Err}(x) &= \mathrm{E}[(f(x) - f^*(x))^2] = \\
&= \mathrm{E}[f(x)^2 - 2f(x)^* f(x) + f^*(x)^2] = \\
&= \mathrm{E}[f(x)^2] - \mathrm{E}[2f(x)^* f(x)] + \mathrm{E}[f^*(x)^2] = \\
&= \mathrm{Var}[f(x)] + \mathrm{E}[f(x)]^2 + \mathrm{Var}[f^*(x)] + \mathrm{E}[f^*(x)]^2 - 2f(x)\mathrm{E}[f^*(x)] = \\
&= \mathrm{Var}[f(x)] + \mathrm{Var}[f^*(x)] + (f(x)^2 - 2f(x)\mathrm{E}[f^*(x)] + \mathrm{E}[f^*(x)]^2) = \\
&= \mathrm{Var}[f(x)] + \mathrm{Var}[f^*(x)] + (f(x) - \mathrm{E}[f^*(x)])^2 = \\
&= \mathrm{Var}[f(x)] + \mathrm{Var}[f^*(x)] + \mathrm{E}[f(x) - f^*(x)] = \\
&= \sigma^2 + \mathrm{Var}[f^*(x)] + \mathrm{Bias}[f^*(x)]^2
\end{aligned} \tag{2.46}$$

where the first therm is the noise term and cannot be reduced by any model. Hypothetically, a true model with infinite data would be able to reduce both the bias and variance terms to zero, however with imperfect models and finite data there is a trade-off when minimizing both terms.

In practice the squared bias of any model can be infinitely decreased towards zero by adding model complexity by using a more flexible function or increasing the dimension of the input space. However this practice increases variance greatly which results in overfitting the data. Despite that, models with little to no complexity tend to underfit the data because they have high bias and low variance. The relationship of bias and variance between model complexity is demonstrated in figure 2.13 [111] and it exhibits an area where the bias and variance curves intersect, which represents optimal model complexity balancing both types of errors [70].

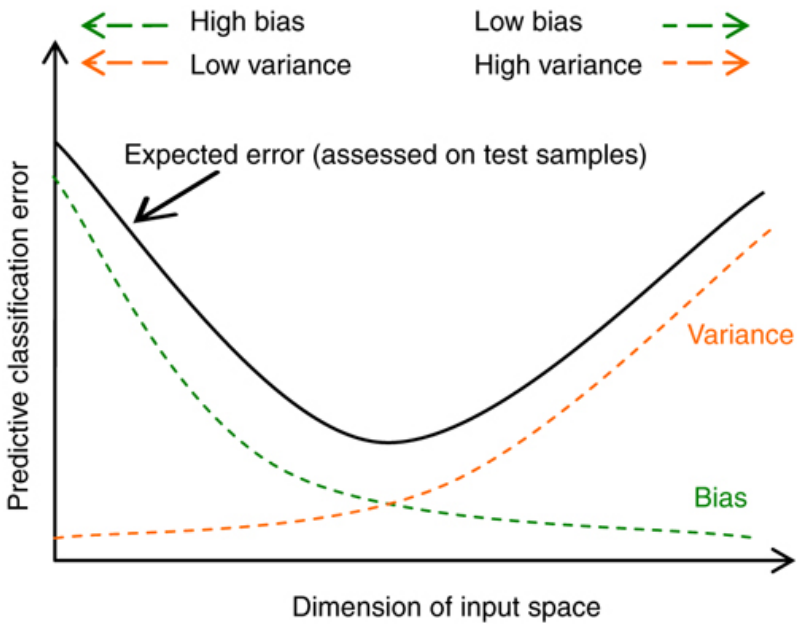


Figure 2.13: Graphical representation of the bias-variance dilemma. Image taken from reference [70]

Regularization algorithms such as ridge regression and LASSO, which are about to be discussed, are designed to optimize model complexity, using these notions about bias and variance related errors.

Ridge regression: Also known as weight decay, it is useful to address ill posed problems where traditional regression methods fail to deliver a solution or, instead, deliver more than one [112].

Imagining that there exists an independent variable matrix, $\mathbf{X}$, and a dependent variable vector, $\boldsymbol{y}$, the traditional regression approach would be to find the values of a vector coefficient, $\boldsymbol{\beta}$=$\beta_i, \beta_2, ..., \beta_i$, such as [70]:

$$\mathbf{X}\boldsymbol{\beta} = \boldsymbol{y} \tag{2.47}$$

However, for an ill posed problem no proper vector $\boldsymbol{\beta}$ would be given, so the end result would be an overdetermined or underdetermined system of equations which would imply that the model would either overfit or underfit new data respectively, according to the bias-variance dilemma discussed earlier. This happens mainly when the several dependent variables are highly correlated with each other which makes

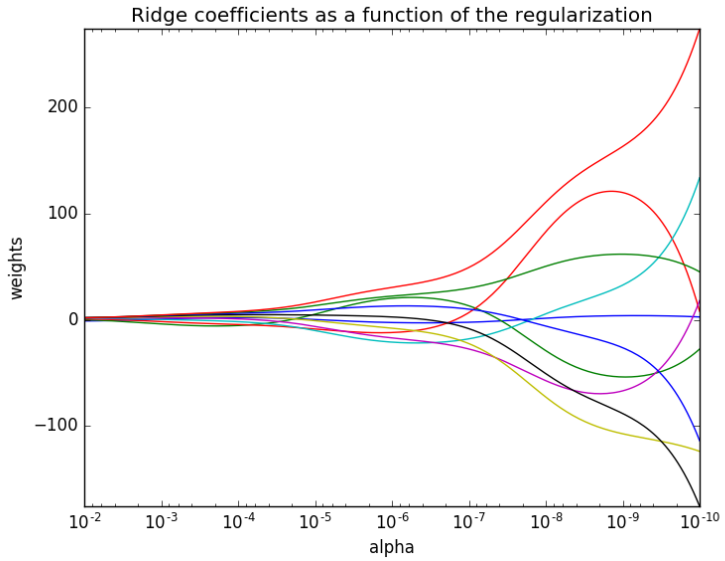


Figure 2.14: Ridge coefficients as a function of the regularization parameter α. Image taken from reference [73]

it so that the values of β_i are either all very large or very small which introduces a large amount of variance into the model [73].

In order to give preference to a particular solution, to increase bias and reduce variance, a regularization term can be added to the least squares minimization process which constrains the size that each β_i can have. This is done by adding an L_2-norm[4] [113] vector term to the least squares minimization process:

$$||\mathrm{X}\beta - y||_2^2 + ||\Gamma\beta||_2^2 = 0 \tag{2.48}$$

where Γ represents the Tikhonov matrix; usually a multiple of the identity matrix such as:

$$\Gamma = \alpha \mathrm{I} \tag{2.49}$$

The addition of the regularization matrix, in the form of the hyper-parameter α, forces the values of β during the least squares minimization to obey the constraint

[4]The length of a vector according to the L_2-norm is given by: $||x||_2 = \sqrt{x_1^2 + \cdots + x_n^2}$

of:

$$||\beta||_2^2 \geq c^2 \tag{2.50}$$

The geometry is demonstrated in figure 2.15 for an hypothetical 2D case and where c is just a constant value imposed by the α parameter. The level of curves of the function must intersect with the circle to get the respective β values. Hence, the minimization for an i^{th}-dimension problem becomes:

$$F(\beta_o, \beta_1, \beta_2, ..., \beta_i, \alpha) = [\sum_n (y_n - \beta_0 - \beta_1 x_n - \beta_2 x_i - ... - \beta_i x_i)^2 \\ +\alpha(\beta_0^2 + \beta_1^2 + \beta_2^2 + ... + \beta_i^2 - c^2)] \tag{2.51}$$

The analytical solution for this problem can be achieved by writing all the partial derivatives of the minimization function and setting them to zero. Solving the resulting system of equations would yield the optimal values for the parameters including the α value. However, there is also a numerical solution which is obtained by iterating the function for several tentative values of α. For each α the β values are calculated along with the respective coefficient of determination. The process continues until a satisfactory value for the coefficient of determination is found by [70, 73]:

$$\min_\beta = ||\mathrm{X}\beta - \mathrm{y}||_2^2 + \alpha||\mathrm{I}\beta||_2^2 \tag{2.52}$$

and for the optimal α value the least squares solution can be written as:

$$\beta = (\mathrm{X}^\mathrm{T} + \alpha\mathrm{I})^{-1}\mathrm{X}^\mathrm{T}y \tag{2.53}$$

Figure 2.14 shows the relationship between the size of α and the β parameters. As its value increases towards a large number all parameter values tend towards zero in order to solve the minimization problem, equation 2.52, causing overfitting. On the other hand, if α decreases in direction of zero, the parameter values will close in to their linear regression counterparts creating an underfitting situation [70, 73].

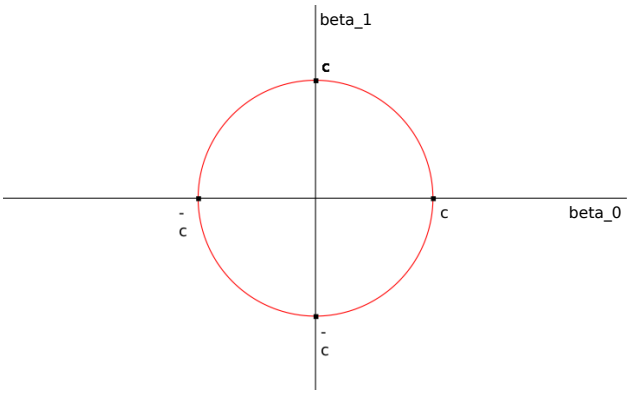


Figure 2.15: Geometry behind the additional constraint imposed bu the ridge regression method.

Least absolute shrinkage and selection operator: Called of LASSO for simplicity's, this method is similar to the ridge regression approach as it is considered a regularization technique, however it is also capable of performing variable selection. [70, 114]

The mathematical formulations of the LASSO regression are almost identical in every way to those of the ridge regression, despite one small key difference: the regularization term is an L_1-norm[5] [113] vector:

$$||\mathrm{X}\beta - y||_2^2 + ||\Gamma\beta||_1^2 = 0 \tag{2.54}$$

which changes the geometry of the optimization problem. Considering again, an hypothetical 2D space, instead of a circle, the constraint takes the form of a diamond like the one represented in figure 2.16. Implying that the minimization problem:

$$\min_\beta = ||\mathrm{X}\beta - \mathrm{y}||_{2^2} + \alpha||\mathrm{I}\beta||_{1^2} \tag{2.55}$$

yielding a much greater change of several coefficients having a zero value. The level curves of the function are much more likely to intersect with the corners of the diamond than with its edges.

One drawback of the LASSO is that there is no analytical solution for finding the coefficients even if an α value is given, it has to be solved numerically [73]. However, this is balanced by the fact that it can exclude unimportant features from the final model. One final note about the LASSO is that all the considerations about the

[5]The length of a vector according to the L_1-norm is given by: $||x||_1 = |x_1| + \cdots + |x_n|$

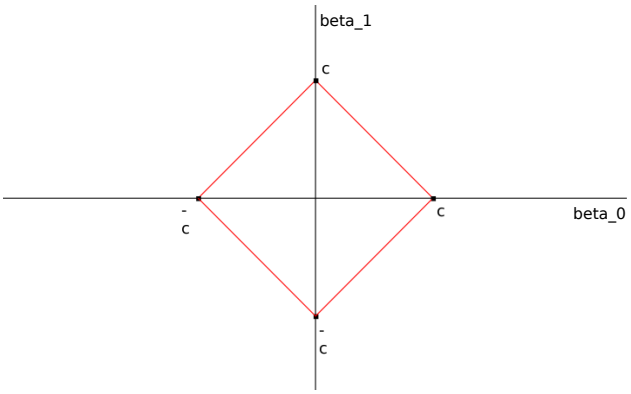


Figure 2.16: Geometry behind the additional constraint imposed bu the LASSO regression method.

relationship between the α value and the parameters given for ridge regression still hold with the exception of more of them being equal to zero.

2.3.2.2 Performance metrics

The last steps of any supervised learning problem include validating the model and measure its performance. The validation step usually includes a cross-validation process and the accuracy of the testing phase is evaluated by a measure of error, such as the mean spared percentage error (MAPE) [73].

Cross-validation: A model validation technique used for assessing how a model will perform when used on new data [115, 116]. In a machine learning scenario using cross-validation consists in splitting the training set into several subsets and using several combinations of them for training and validation [117]. The model is validated if the performance metrics for each training iteration while using all subset combinations are in agreement with each other.

There are several ways of performing the split in cross-validation, however they can all be generalized into the two most common [117]:

- **Leave-p-out cross-validation:** and exhaustive form of cross-validation, it uses p observations to build the validation set and assigns all other to the training set. This process is repeated for every possible way of splitting the original data into a validation set containing p observations and a training set [73].

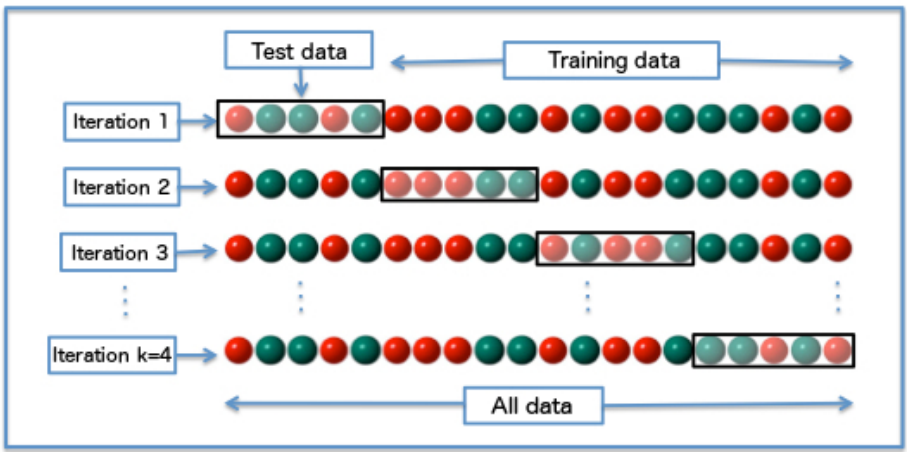


Figure 2.17: Visual representation of a k-fold cross-validation with $k = 4$. Image by Fabian Flöck - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=51562781

- **k-fold cross-validation:** A non-exhaustive approach to cross-validation, it divides the original sample data into k equal sized subsets, visually presented in figure 2.17. Out of these subsets a single one is placed aside for validation and the remaining are used collectively as training data. For each iteration the validation subset is switched [73, 118]

For both these cases, after all iterations are concluded and if the metrics of every iteration are in agreement with each other, the model can be used for testing in one of two ways [119]:

- **Standard cross-validation:** the results of each iteration are averaged and used for building the final model prior to testing.
- **Winner-takes-all cross-validation:** The results of the best iteration are directly used in the final model for testing.

While a leave-p-out cross-validation is more reliable and produces more accurate models than the k-fold cross-validation it is also computationally much more demanding. The same applies for the standard and winner-takes all cross-validations, since the standard is more time and resource demanding but in some cases it generates more precise models. However, as an approximation, the less demanding combination (winner-takes-all k-fold cross-validation) is shown to be sufficient for most cases which makes the more demanding combination (leave-p-out standard cross-validation) excessive in most situations [73, 115, 116, 119].

Mean absolute percentage error: Is the standardized approach to measure prediction accuracy of forecasting methods in statistics and machine learning theory [120]. It is defined by the formula [119]:

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \tag{2.56}$$

that translates MAPE as percentage based on the absolute difference between the actual values, A_t, and the corresponding forecasted values, F_t.

For comparison using the same method with different data sets, MAPE provides a simple and efficient way of measuring accuracy. However its limitations should be know as it fails to compare different models since it turns biased towards models that mostly forecast values lower than their real counterparts. Beyond that, it also cannot be used for zero values, $A_t = 0$, and it has no upper limit for errors of individual predictions that overestimate the real value [121].

2.3.3 Unsupervised learning methods

When dealing with unlabeled data, that does not have an explicit target variable included in the observations, it is not possible to use any of the learning methods discussed so far since they require a dependent variable to be given. However, unsupervised machine learning methods are able to infer a function to describe hidden patterns and structures in data without an explicit target variable which makes them appealing since they can be used on data structures without requiring the user to have any knowledge about them. Despite that, they are not without limitations because since the training examples are unlabeled there are no means to evaluate model accuracy and performance [70].

Traditionally, these methods are divided into two separate categories, as shown in figure 2.18: clustering [122] and dimensionality reduction [123]. *Clustering* is the unsupervised learning version of classification algorithms while *dimensionality reduction* is used for feature extraction and data analysis, being equivalent regression models, specifically to regularization algorithms.

Cluster analysis is tasked with grouping objects in the same groups (called clusters) in a way that objects placed together are similar in at least some way. It attributes a class, or label, to objects, that was hidden or unknown to the user. This

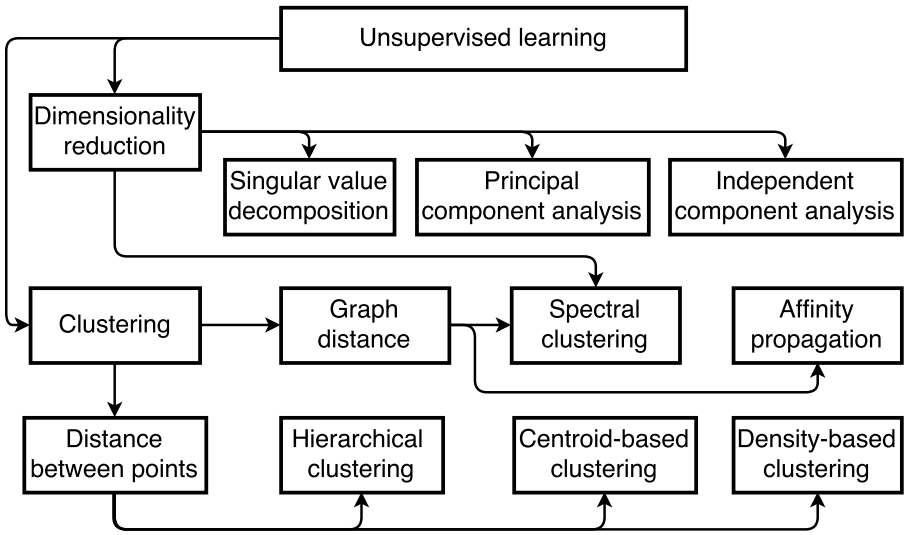


Figure 2.18: Most commonly used unsupervised learning algorithms.

makes clustering highly effective as a data mining tool, being capable of executing tasks such as pattern recognition, image analysis and information retrieval. There are several algorithms for performing cluster analysis: those based on measuring distances between points, such as hierarchical clustering, centroid-based clustering and density-based clustering; and those based on measuring distances between graphs, like affinity propagation and spectral clustering [122].

Dimensionality reduction aims at reducing the number of random variables under consideration by filtering out all the unnecessary variables while retaining all the important ones improving model simplicity, efficiency, accuracy and performance. Most algorithms belonging to this class are considered some type of factor analysis, and include principal component analysis (PCA), independent component analysis (ICA) and singular value decomposition (SVM) [124]. Spectral clustering can also be considered a dimensionality reduction method even though it is primarily a clustering method [125].

Centroid-based clustering: Called of k-means clustering, is a vector quantization method and one of the most widely used clustering techniques for data mining. It aims at splitting n data points into a k number of clusters, C, in a way that each

group has equal variance, minimizing inertia [73]:

$$\text{Inertia} = \frac{\text{Intragroup sum of squares}}{\text{Total sum of squares}} \tag{2.57}$$

However it requires the number of clusters, k, to be specified. Each cluster is described by a mean, μ_j, of the samples in it, which is called a *cluster centroid*. These centroids are not data points and are responsible for defining the inertia. For a given number of clusters, k, finding the optimal centroids in order to minimize inertia is the essence of the method [73, 126]:

$$\text{Intragroup sum of squares} = \sum_{i=0}^{n} \min_{\mu_j \in C}(||x_j - \mu_j||^2) \tag{2.58}$$

Giving a measure of how internally coherent the clusters are. Despite that, it has a few drawbacks: it assumes that clusters are convex and isotropic, so it fails to properly describe elongated or oddly shaped clusters and it is not normalized, so for very high dimension problems it is afflicted by the *curse of dimensionality*[6] [73] because the distances tend to become inflated.

Regardless, the inertia is always constricted by the number of clusters, meaning that if a "wrong" number of clusters is chosen model performance will fail. A common methodology for finding the optimal number of clusters is the elbow method [127]. It relies on running the k-means clustering algorithm for the same problem using an increasingly number of clusters in each iteration and measuring its inertia, as a measure of the explained variance.

Plotting the explained variance versus the number of clusters generates a graphic, like the one in figure 2.19, where the increase in explained variance is very steep in the begging but it reaches a point where it begins to stabilize, known as the "elbow". Adding clusters beyond this point will not result in a significant increase of modeling capability to justify the increase in complexity. Though this point is not always very explicit, but is can be more easily found by plotting the second derivative of a fitting function whose maximum identifies the "elbow" [127].

[6] As the dimensionality increases, the volume of the space increases much more rapidly which causes available data to become insufficient to describe it. This compromises the statistical significance of the results.

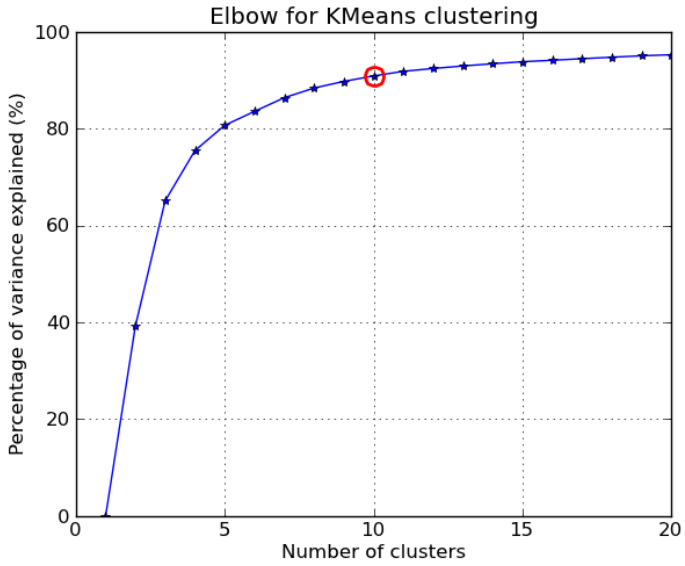


Figure 2.19: Explained variance versus th number of clusters for a k-means clustering algorithm. The circle at $k = 10$ indicates the "elbow point".

Hierarchical clustering: The main advantage of the models lies in the fact that there is no need to specify a specific number of clusters beforehand. Also known as connectivity-based clustering, they state that objects are more related with those that are nearby than farther away but are still related via a proximity based hierarchy [73].

Creating that hierarchy of clusters based on the proximity of the data points is how these models work. And that can be built in one of two ways: using a *divisive approach* or an *agglomerative approach* [128].

In the divisive approach, also known as the "top down" approach, a k-mean algorithm is used for an initial assessment creating a specified number of clusters, $c_i : i = 1 \cdots k$, and for each cluster created the k-means algorithm is applied again continuing to further divide the clusters. This process can run until it exists a cluster per data point (singleton clusters) producing a hierarchy. This approach is very fast and efficient and has low complexity; Each iteration operates on a slice [128]:

$$\mathcal{O}(\mathrm{knd} \cdot log_{\mathrm{k}}\mathrm{n}) \tag{2.59}$$

and k is the specified number of clusters, n is the number of data points and d is the dimensionality of the data. Assuming that each iteration splits the cluster

into equal sized sub-clusters the level of computation remains the same along the process. However it is also very greedy, meaning that once a data point is assigned to a certain cluster it cannot be compared with data points belonging to other clusters even if they are nearby in the original data.

The agglomerative approach, known as "bottom up" approach, does the opposite to try and ensure that nearby points end up in the same cluster. Instead of starting with large clusters and splitting them, it starts with a collection of n singleton clusters, C, where each cluster contains only one data point, $c_i = \{x_i\}$, and proceeds to iteratively merge them together [128]:

1. Find a pair of clusters that is closest: $\min_{i,j} D(c_i, c_j)$;
2. Merge the clusters c_i, c_j into a new cluster c_{i+j};
3. Remove c_i, c_j from the collection C and add $c_{i,j}$ to it;
4. Check if number of clusters is bigger than one:
   (a) if $C \geq 1$: repeat from step 1;
   (b) if $C = 1$: terminate algorithm.

Although nearby data points have a greater chance of ending up in the same cluster when compared with the divisive approach, making it less greedy, model complexity is much greater [128]:

$$\mathcal{O}(\mathrm{n}^2\mathrm{d} + \mathrm{n}^3) \tag{2.60}$$

Since the distance between every singleton has to be computed once at the beginning, n^2d, and in each iteration it has to be traversed in order to find the closest pairs, n^3. In chemistry hierarchical clustering is used for identifying the largest substructures shared by several molecular structures helping scientists to quickly find novel examples of active compound families, as presented in figure 2.20 [129, 130].

Density-based clustering: Areas of higher density in the data set are defined as clusters and are separated by areas of lower density [131]. Objects in these lower density areas are classified as either noise or outliers. This criteria makes it possible to have clusters of any shape. Recently this method has been used in data mining and analysis of high-dimensional images take by X-ray based spectro-microscopy [132].

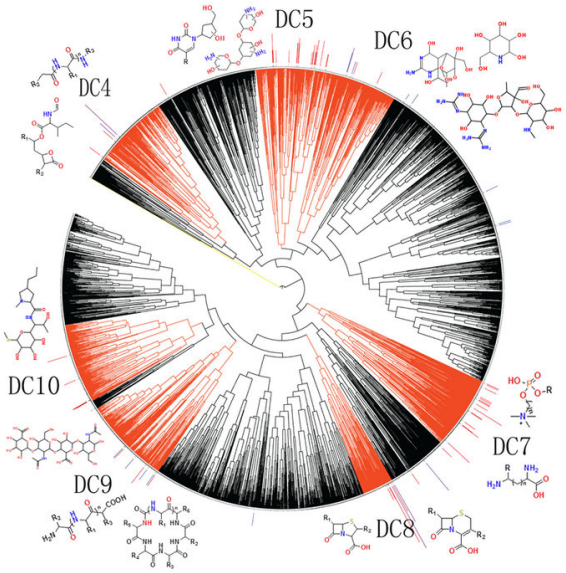


Figure 2.20: Example of a clustered distribution of drugs. Image taken from reference [130].

Scientifically, the most popular density-based clustering method is called DBSCAN (density-based spatial clustering of applications with noise) [133]. It classifies data points into three different categories, as denoted in figure 2.21: *core points*, *reachable points* and *outliers*; according to two parameters: the minimum number of neighboring samples, $MinPts$, and a distance value, eps. Based on these to parameters, the data points classification is done as [133]:

- **core point:** within the distance of eps there exists and equal or bigger number of data points than the minimum number of neighboring samples: neighbors in $eps \geq MinPts$;
- **reachable points**: are neighbors of a core point but have insufficient neighbors of their own to be considered as core points;
- **outliers**: are not neighbors of core points nor have sufficient neighbors of their own to be considered core points;

All points within a cluster are mutually connected by density. By definition, both core and reachable points are always part of a cluster while outliers are unassigned.

The algorithm starts by picking an arbitrary point from the data set, then retrieves and evaluates its neighborhood according to the chosen parameters. If it passes, a cluster is started and the neighboring points are evaluated in the same

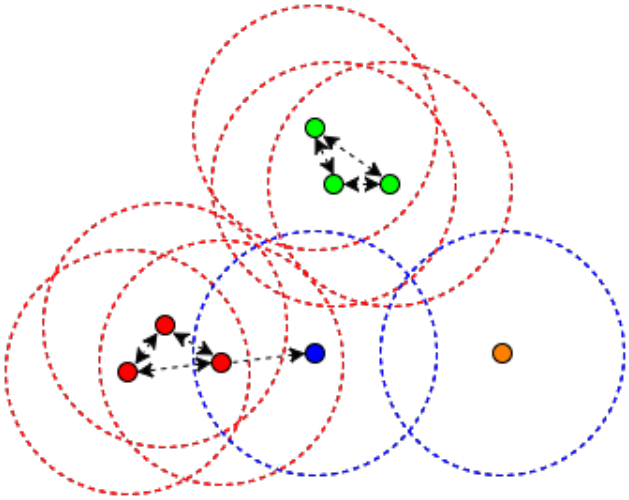


Figure 2.21: DBSCAN clustering illustration. Red and green points are the core points of two separate clusters, the blue point is a reachable point belonging to the red cluster and the orange point is an outlier. Image retrieved from the "Machine Lwearning notebook" (https://sites.google.com/site/machinelearningnotebook2/) on May, 22nd of 2017.

way to expand it until only reachable points are detected in which case the cluster is finished. On the other hand, if the starting point fails to meet the parameters it is immediately deemed an outlier, however this does not exclude it from being part of the neighborhood of another core or reachable point so it can still be made part of a cluster. The algorithm ends after each data point has been assigned a specific and permanent role as a core, reachable or outlier point [73, 133].

The advantages of density-based clustering lie in the fact that it does not require a pre-specified number of clusters to be given, that it can find oddly-shaped clusters, that it has a notion of noise and robustness and that it requires minimal parameter input. However, since it starts by arbitrarily choosing a starting point it is not deterministic because border points might end up in different clusters depending on the starting order, its overall quality depends on the input distance, *eps*, so for high-dimensional data it suffers from the *curse of dimensionality* and it cannot be used on data sets with large differences in densities [73].

Affinity propagation: Based on "massage passing" between data points, it explores the idea that data points have a bigger *affinity* with a few other points and uses that concept to find *exemplars*: objects of the original data set that are representative of clusters [134].

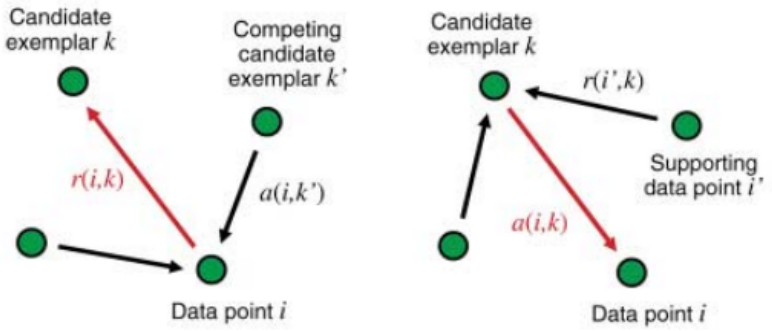


Figure 2.22: Illustration of the message exchange procedure for an affinity propagation clustering algorithm. Image taken from reference [134].

The input is a collection of the *similarities* between data points, $s(i,k)$, that translates how well the data point k is suited to be an exemplar for data point i. For an optimization process, the similarity function is defined as the negative squared error of the two indexes [134]:

$$s(i,k) = -||x_i - x_k||^2 \tag{2.61}$$

It does not require a specified number of clusters to be given, because the real-valued (self-)*preference* functions, $s(k,k)$, are specified in the input and larger preferences values are more likely to become exemplars. The number of clusters is influenced by these values but is refined by the *message-passing procedure*, illustrated in figure 2.22 [134], that takes place afterwards. Messages exchanged between data points are of two types: *responsibility*, $r(i,k)$,and *availability*, $a(i,k)$. Both are in competition with each other and can be combined during any stage of the algorithm runtime to decide which points are exemplars and, for all that aren't, to which exemplar they belong to [134].

Responsibility pertains to the messages that are sent from data point i to its candidate exemplar k indicating how well-suited k is to serve as an exemplar to i while still accounting for all other possible exemplars for it. Availability, on the other hand, relates to the messages sent from the candidate exemplar k to point i reflecting how appropriate it would be for point i to have point k as its exemplar while accounting for other points that are also considering k as its candidate exemplar [134].

The algorithm starts by creating the responsibility, $\mathbf{R}$, and availability, $\mathbf{A}$, matrices and initializing them to all zeros. Then it proceeds to calculating the responsibility and availability functions iteratively. First starting with the responsibility [73, 134]:

$$\begin{aligned} r(i,k) &= s(i,k) - \max_{k' \text{ s.t. } k' \neq k}\{a(i,k') + s(i,k')\} = \\ &= s(i,k) - \max_{k' \text{ s.t. } k' \neq k}\{s(i,k')\} \end{aligned} \tag{2.62}$$

And using it to update the availability:

$$\begin{aligned} a(i,k) &= \min_{i \neq k}\left(0, r(k,k) + \sum_{i' \notin \{i,k\}} \max(0, r(i',k))\right) \\ &= \sum_{i' \neq k} \max(0, r(i',k)) \end{aligned} \tag{2.63}$$

The calculation ends when the cluster boundaries remain the same for a pre-specified number of iterations or until a pre-determined maximum number of iterations is reached. The exemplars are then identified in the matrices as those whose cumulative values of responsibility and availability are positive:

$$r(i,i) + a(i,i) \geq 0 \tag{2.64}$$

and then are extracted to determine the number, center and positions of the clusters.

Though it is a very reliable clustering algorithm and requires no initial number of clusters to be given, it is very complex. Its time complexity is of order [73, 134]:

$$\mathcal{O}(\mathrm{N}^2\mathrm{T}) \tag{2.65}$$

Given an N number of data points and a T number of iterations until convergence. And its memory requirements are also very large:

$$\mathcal{O}(\mathrm{N}^2) \tag{2.66}$$

Due to the construction of the responsibility and availability matrices required to perform the calculation. Given these constraints, it is a method best used on small

to medium sized data sets. It finds its use in chemistry in the realm of computational biology for modeling molecular systems in biological systems [73, 134].

Spectral clustering: Is an hybrid between clustering and dimensionality reduction techniques as it performs a low-dimension embedding of the affinity matrix creating a reduced-dimension clustering problem that can be addressed by a conventional clustering algorithm such as k-means [135].

The first step consists in building an affinity matrix to measure the similarities between the several objects. Although there is no singular formal way to define such a matrix, usually they are created using the inverse of some distance metric, such as the Euclidean distance [135]:

$$s(x, y) = -||x - y||_2^2 \tag{2.67}$$

Meaning that similar objects are represented by large values while small values indicate the presence of dissimilar objects.

The next step consists performing the eigendecomposition of the affinity matrix to find its canonical form in terms of its eigenvalues and eigenvectors. Note that the nature of the construction of the affinity matrix makes it diagonalizable. Considering an affinity matrix, $\mathbf{A}$, a non-singular eigenvector matrix, $\mathbf{P}$, and a diagonal eigenvalue matrix, $\mathbf{D}$ [136]:

$$\mathbf{A} = \mathbf{PDP^{-1}} \Rightarrow \tag{2.68}$$

$$\begin{aligned} \mathbf{A^2} &= (\mathbf{PDP^{-1}})(\mathbf{PDP^{-1}}) = \\ &= \mathbf{PD}(\mathbf{P^{-1}P})\mathbf{DP^{-1}} = \\ &= \mathbf{PD^2P^{-1}} \end{aligned} \tag{2.69}$$

Extrapolating for general positive powers, equation 2.69 becomes:

$$\mathbf{A^n} = \mathbf{PD^nP^{-1}} \tag{2.70}$$

Also, by calculating the inverse of $\mathbf{A}$ one can see that the same results hold for

negative powers as well:

$$\begin{aligned}\mathbf{A^{-1}} &= (\mathbf{PDP^{-1}})^{-1} \\ &= \mathbf{PD^{-1}P^{-1}}\end{aligned} \tag{2.71}$$

Given that equation 2.70 holds of positive and negative values of the power, and that $\mathbf{A}$ is diagonalizable by definition, eigendecomposition of the affinity matrix is always possible. After the decomposition, eigenvectors whose eigenvalue is small are discarded as they are assumed to represent noise while vectors with large eigenvalues are used to map the system to a new reduced-dimension space that can be dealt with using conventional clustering methods. This also means that the new space has a dimensionality equal to the number of large eigenvalues from the original affinity matrix.

Principal component analysis: A dimensionality reduction method based on a statistical operation that converts a given set of observable variables with an unknown degree of correlation into a set of linearly uncorrelated variables by using an orthogonal transformation [137].

The idea of principal component analysis (PCA) is to fit an n-dimensional ellipsoid (hyperellipsoid) to the data such that each of its axis represents a variable of the data set. Small axis have small variance so they can be discarded from the data representation since the amount of information that is lost in the process is substantially small. Finding the axis and their variances is done by [138]:

1. Subtracting the mean of each variable from the data set to center the data;

2. Computing the covariance matrix of the data:

$$\mathbf{C} = \mathbf{X^T X} \tag{2.72}$$

   And calculating its eigenvalues, $\mathbf{D}$, and eigenvectors, $\mathbf{P}$:

$$\mathbf{C} = \mathbf{PDP^{-1}} \tag{2.73}$$

3. Orthonormalizing the eigenvectors. Considering the following projection operator $\text{proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{v},\mathbf{u} \rangle}{\langle \mathbf{u},\mathbf{u} \rangle}\mathbf{u}$, that projects vector $\mathbf{v}$ orthogonally into the line

spanned by vector $\mathbf{u}$, and $\langle \mathbf{v}, \mathbf{u} \rangle$ as the inner product of the vectors $\mathbf{v}$ and $\mathbf{u}$, the normalized eigenvectors, $\mathbf{e}$, are calculated as:

$$\mathbf{u}_k = \mathbf{v}_k - \sum_{j=1}^{k-1} \text{proj}_{\mathbf{u}j}(\mathbf{v}_k) \text{ so that } \mathbf{e}_k = \frac{\mathbf{u}_k}{||\mathbf{u}_k||} \tag{2.74}$$

4. Calculating the proportion of the variance for each orthonormal eigenvector by diving their individual eigenvalues by the sum of all eigenvalues.

This procedure allows for an efficient way to reduce the dimensionality of high-dimensional data without neglecting much of its original information, however it is highly sensitive regarding data scaling and there is no formal agreement on how to best do it in order to optimize results [139].

Independent component analysis: Contrary to PCA that tries to maximize variance to reconstruct the feature space, independent component analysis (ICA) is tasked with finding a linear transformation able to convert the feature space into one where all variables are independent from each other. Assuming that the observables, in the form of vector $\mathbf{X}$, are caused by some linear combination of other hidden variables, in the form o vector $\mathbf{Y}$, this can be mathematically presented as [140]:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_n \end{bmatrix} = \mathbf{Y} \tag{2.75}$$

Hidden variables are independent from each other, meaning that they don't share any information with each other:

$$\text{I}(y_i, y_j) = 0 \tag{2.76}$$

However, in an ideal case, their collective information should be the same as that presented in the observables. For a non-ideal situation, their shared information,

based on ICA, should be as high as possible:

$$\mathrm{I}(y_i, y_j) = \uparrow \tag{2.77}$$

ICA tries to balance these two aspects when transforming the feature space: the overall shared information should be conserved to a maximum during the transformation, while information shared between hidden variables should be equal to zero. This implies that each of the new dimensions is mutually exclusive but the feature spaces are not [141].

In chemistry, independent component analysis, alongside PCA, is used in signal processing in the realm of analytical chemistry for treating data and analyzing results. [142].

Single value decomposition: Given a real input matrix, $\mathbf{M}_{m\times n}$ there exists a factorization of $\mathbf{M}$ with the form:

$$\mathbf{M_{m\times n}} = \mathbf{U_{m\times r}}\mathbf{\Sigma_{r\times r}}(\mathbf{V_{n\times r}})^{\mathbf{T}} \tag{2.78}$$

In which, $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vector matrices and $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values of $\mathbf{M}$ as the diagonal entries in descending order. This matrix is uniquely determined by the input matrix [143].

Assuming that the the input matrix relates two different variables (m and n) trough its matrix value, the singular value decomposition is able to separate both variables into concepts (r) by means of the left singular vector matrix $\mathbf{U}$ (variable m-to-concept similarity matrix) and the right singular vector matrix $\mathbf{V}$ (variable n-to-concept similarity matrix) and the singular values in the matrix Σ express the strength of each concept in representing its chosen instances.

This method allows to find similar individuals in a population according to chosen parameters and group them together to create a reduced dimensional space while retaining information about the representation of the original population [144]. It has several applications in natural sciences, excelling at genomic research [145].

Experimental section

This study relies on official satellite data, taken over the city of Coimbra, Portugal. Data was provided by the European Space Agency (ESA) under the Copernicus - The European Earth Observation Programme, hence it is considered as empirically correct. The central point for all measurements is the "Instituto Pedro Nunes" (IPN) located at 40.192169N -8.414162W. Because the data was retrieved via satellite and directly supplied by ESA, no consideration was given to the geography and micro-climate of the city itself. In addition, measurements were done on an hourly basis from 01-10-2016 up until 30-09-2017. Which means that a total of 8760 data points were retrieved for each of the six compounds of interest.

3.1 Method selection

Proper selection of a machine learning algorithm for developing the forecasting model required knowledge about the data and a defined goal: collected data is comprised of time-stamped airborne pollutant concentrations over the city of Coimbra, and the goal is to be able to forecast future concentrations. Since the target variables are part of the observations and are comprised of numerical values, supervised regression methods are the proper choice for constructing the machine learning forecasting algorithm. This though process is demonstrated in figure 3.1 indicated by the green line.

Out of all the available regression algorithms, the one used for constructing the forecasting machine is based on multiple linear regression. This choice is a rather subjective one and is based on a few considerations about the data and the end-goal which is the creation of a forecasting network over the city.

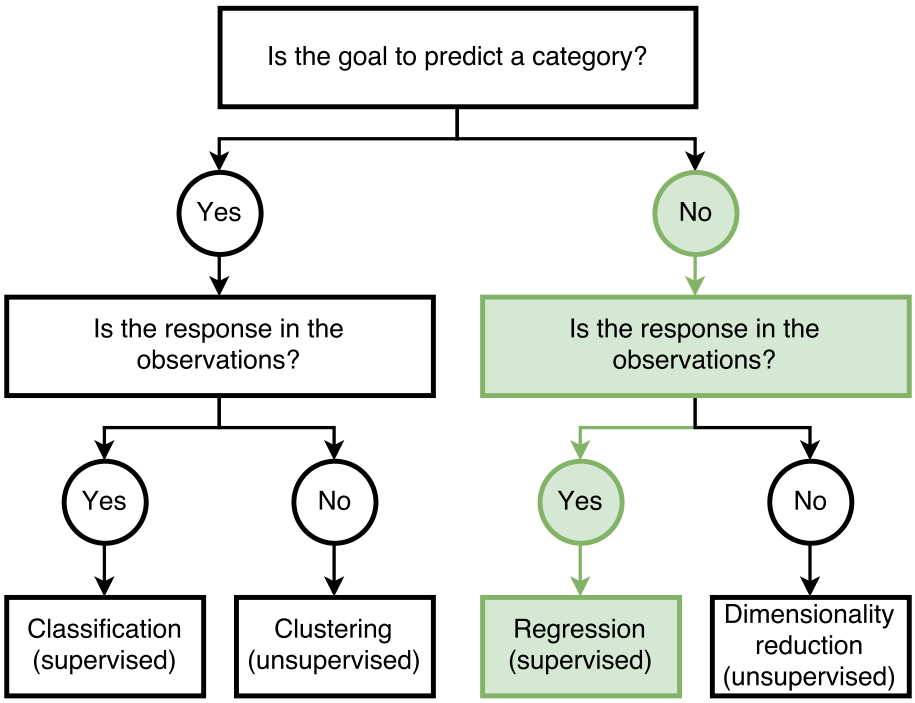


Figure 3.1: Algorithm choice flowchart with the choice process highlighted by the green path.

Firstly, linear regression was opted over non-linear regression because of its simplicity. Since the end-goal is to have an algorithm capable of working near-real time, the simpler it is the better since there is less lag between forecasts. And secondly, because each data set is comprised of two columns (one denoting time and one the respective concentration of a single pollutant), the data is two dimensional, so there is no need to over-complicate by using multivariate or regularization models. Because of this, the choice would be limited to either simple regression or multiple regression, but because the desired forecasting horizon contains more than one variable, it constrains the choice to multiple regression.

3.2 Procedure

The data preparation steps are presented in figure 3.2. Firstly, the data needed to be cleaned and treated, which required sorting all measurements chronologically and calculating properties such as the mean and standard deviation to provide some statistical sense to it. Afterwards it was necessary to verify the existence of data gaps, absent data points from failed measurements, and determine the percentage

of the full data set that was missing. Upon verifying the existence of absent data, the missing data points were inferred by using a simple linear regression estimator and the statistical properties were measured again to make sure that the overall behavior of the time-series did not change significantly.

The second step, after cleaning the data, was to verify if each time-series constructed was stationary, which was done by using the Dickey-Fuller test. And after testing, three copies of each data set were created for filtering and signal smoothing while the original was kept aside for serving as a control reference.

The third step involved applying the two moving average and the Savitzky-Golay filters to the data set copies to smooth them. This resulted in the creation of three new data sets for each compound, which needed to be tested for stationarity again, a process that was performed with the Dickey-Fuller test, the same used on the original data sets. The outcome of this process was a total of four data sets for each pollutant: the original and three different smoothened signals. This amounted to a total of different 24 data sets, meaning that a full 210240 data points existed for training and testing the machine learning algorithms.

3.2.1 Training and testing

After cleaning, treating and smoothing the data sets (or signals) the last step was to effectively train and test a machine learning algorithm. Since time-series are sequential data sets, the first 80% were selected for training and validation and the last 20% were reserved for testing the algorithm.

In order to train and validate each model, the 80% training set was further divided into ten equally-sized sub-sets (called folds) for cross-validation. Afterwards, all possible combinations of nine different folds were considered and used individually for training while leaving the remaining fold in each iteration for testing. The iteration that gave the best results was replicated to produce the final model which was then tested on the remaining 20% of the data set.

In the testing phase, three test sets were created from the 20% test set using a time lag of two days: from the first sub-set the last 48 data points were excluded, from the second sub-set the first and last 24 data points were excluded and from the third sub-set the first 48 data points were excluded. This was done in order to be able to replicate each model three times to achieve statistical meaning and to make

sure that the model was independent of the starting point of the test set.

On each iteration the mean absolute percentage error value was calculated and outputted for model performance analysis. The MAPE values resulting from the three testing iterations were compared against each other to assure that model coherence existed and their mean value was used for comparison between models.

1. **Clean the data:**
   (a) Check percentage of missing data;
   (b) Measure mean and standard deviation;
   (c) Infer missing data points using a linear regression estimator;
   (d) Measure new mean and standard deviation for comparison;
2. **Test for stationarity (Dickey-Fuller Test);**
3. **Filter the data:**
   (a) Using a seven days moving mean:
      - Re-test for stationarity;
   (b) Using a 28 days moving mean:
      - Re-test for stationarity;
   (c) Using a 3^{rd} degree polynomial Savitzky-Golay filter:
      - Re-test for stationarity;
4. **Run the MLA on all data sets;**
   (a) Split the data set into a training set (80%) and test set (20%);
   (b) Train the MLA:
      i. Split the training set into ten equal-sized sub-sets;
      ii. Train and validate the MLA using a ten-fold-winner-takes-all cross validation approach;
   (c) Test the MLA:
      i. Differentiate the test set into three sub-sets;
      ii. Calculate performance metrics for each test sub-set;
      iii. Evaluate model performance.

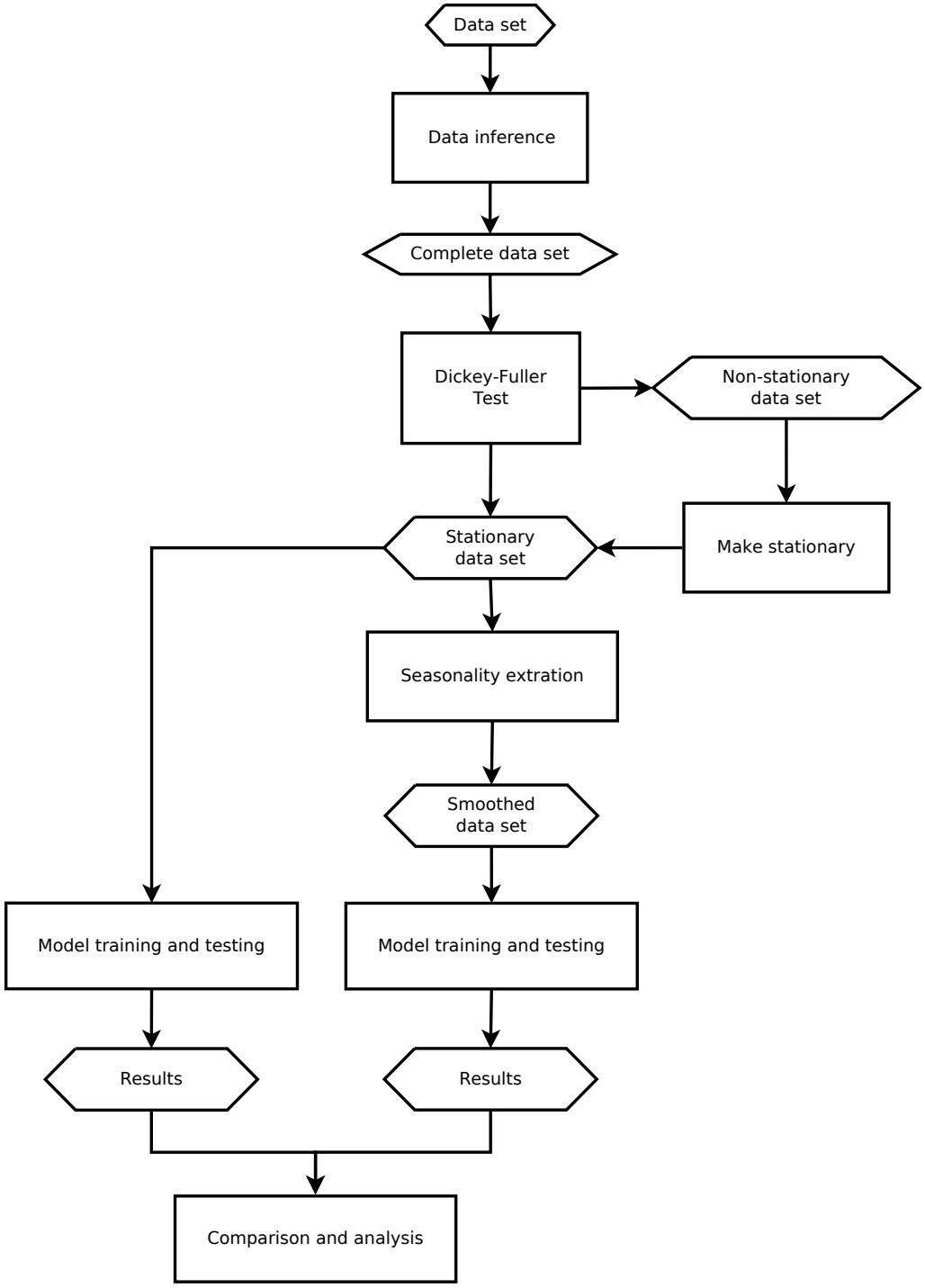


Figure 3.2: Data processing flowchart and algorithm application.

Results and discussion

4.1 Carbon monoxide

Approximately 6.59 % of the carbon monoxide data set was comprised of missing measurements. The remaining 93.41 % of the signal was averaged at $0.314\,\mathrm{ngm}^{-3}$ with a standard deviation of 0.114. After inference, the signal new mean and standard deviation was of $0.336\,\mathrm{ngm}^{-3}$ and 0.122 respectively indicating that signal behavior remained constant after data inference. This meant that the signal could be smoothened without any restraints.

Given the total of 8746 data points that made up the signal (including inferred data) the critical values used for the Dickey-Fuller test were:

- Critical Value (10%) = -2.566946
- Critical Value (5%) = -2.861870
- Critical Value (1%) = -3.431097

Which resulted in the confirmation that every data set, the original and the smoothened signals, were stationary with a confidence level of 99 %. This because the test values, shown in table 4.1, are all inferior to the critical values. In addition, since the p-values for every signal were very close to zero, it indicated that the data contained within each signal was enough to reject the null hypothesis according to the Dickey-Fuller test outcome.

Carbon monoxide measurements came in parts per million which are relatively small numerical values so they were scaled by converting them into $\mathrm{ng/m}^3$ prior to algorithm training. Given that the four signals are stationary, they could be used for model training.

4.1.1 CO model performance

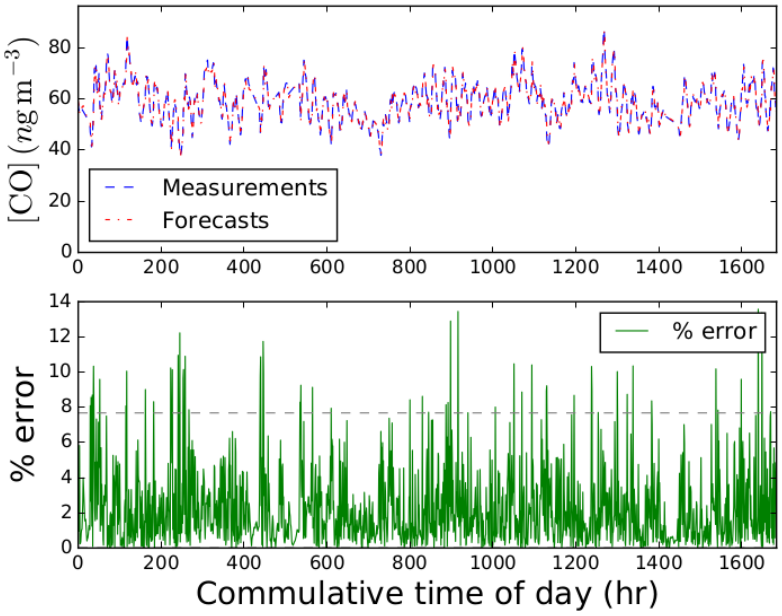
Using the original carbon monoxide signal for training, the machine learning algorithm gave MAPE values of 7.748 %, 7.728 % and 7.680 % for the three testing iterations. These are small and coherent with each other, as supported by the standard deviation of 0.0349, which points towards an accurate model. Analysis of the graphics in figure 4.1, shows that individual errors did not go over 14 % further indicating a viable model. Moreover, the dispersion plot (on the right-handed side) demonstrates that there was a good coherence between forecasted and real values. All these factors indicate a viable model, with its overall performance being given the the form of a mean MAPE of 7.719 %.

Applying the seven day moving average filter on the signal prior to using to train the model yielded individual MAPE values of 9.531 %, 9.946 % and 11.427 % in the testing phase. These values are higher than the previous model and more disperse given the standard deviation of 0.996, however they are still relatively small enough to consider the model viable. ALthough the las testing iteration MAPE seems to be slightly offset-ed from the others. Graphical analysis of the results in figure 4.2, shows that individual error values did not go over 16 % and that correlation between real and forecasts is still fairly good. Model performance is evaluated by the mean MAPE value of 10.301 %.

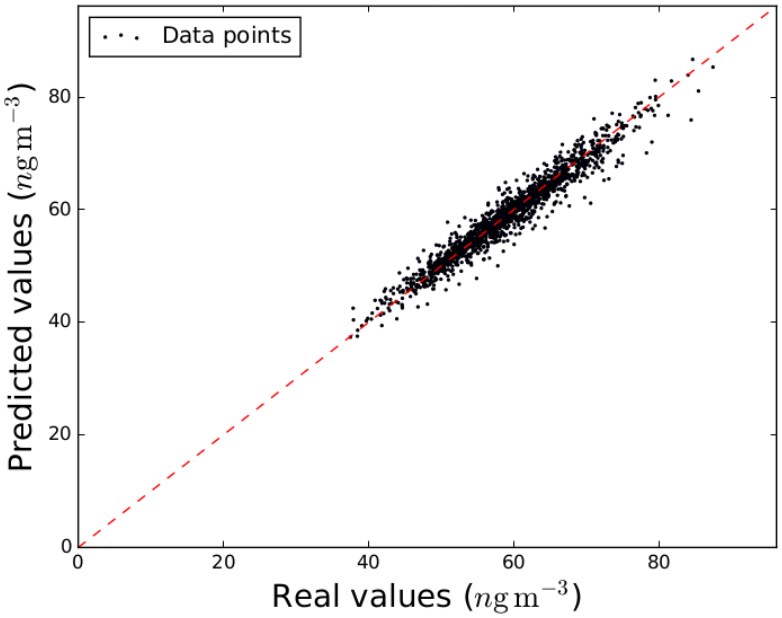
Changing the moving average window from seven to 28 days increased the mean absolute relative errors obtained while testing to 12.658 %, 13.160 % and 13.861 %. This places model performance at the mean MAPE of 13.227 % with a standard deviation of 0.604. Both these values suggest an accurate and consistent model. These assumptions are supported by the analysis of the plots generated by the model, shown in figure 4.3, where it is visible that individual errors barely surpassed the MAPE value and that coherence between forecasts and measurements is present.

Table 4.1: Dickey-Fuller test results for CO signals.

	Observations	Test value	P-value	Stationary
Original	8746	-5.148123	1.10E-05	Yes
7 day filter		-12.91134	4.05E-24	Yes
28 day filter		-9.216728	1.83E-15	Yes
Savitzky-Golay filter		-4.859107	4.20E-05	Yes

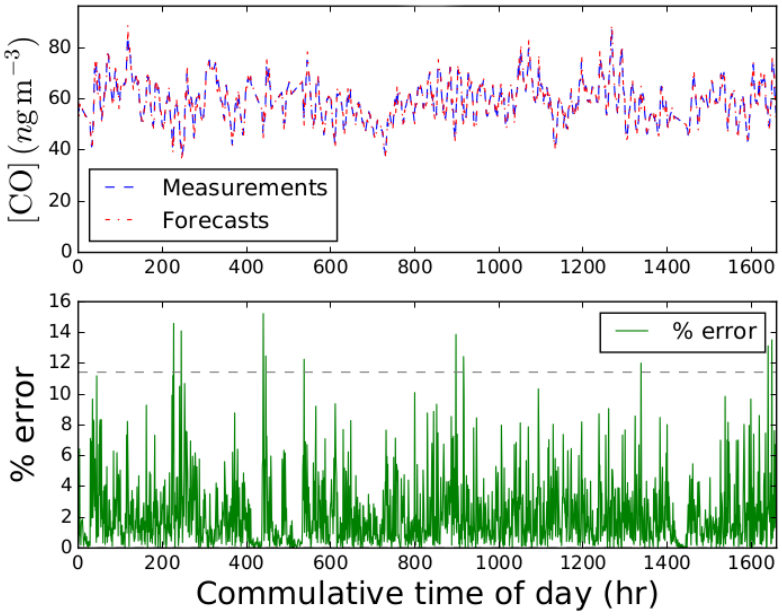


(a)

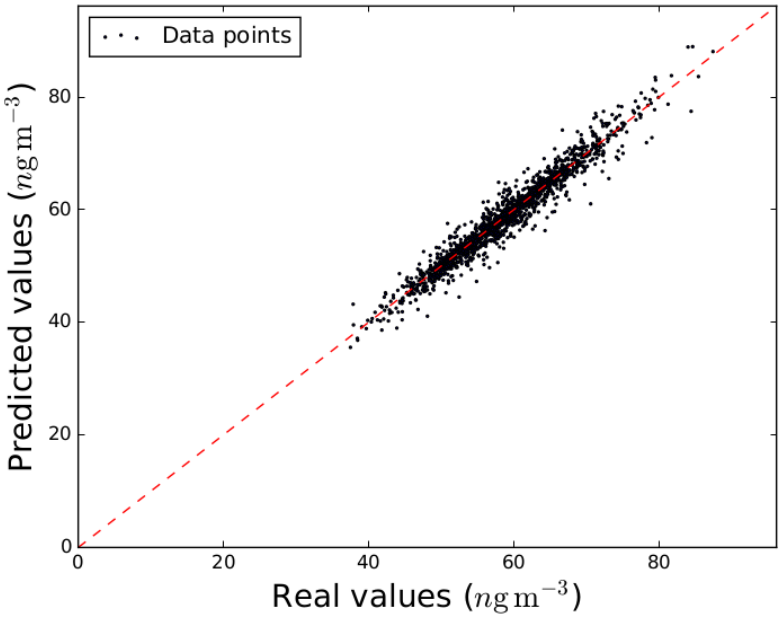


(b)

Figure 4.1: Un-smoothened 24 hour CO forecasting results for the third iteration. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.



(a)



(b)

Figure 4.2: Smoothened 24 hour CO forecasting results for the third iteration using a moving average with a window of seven days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

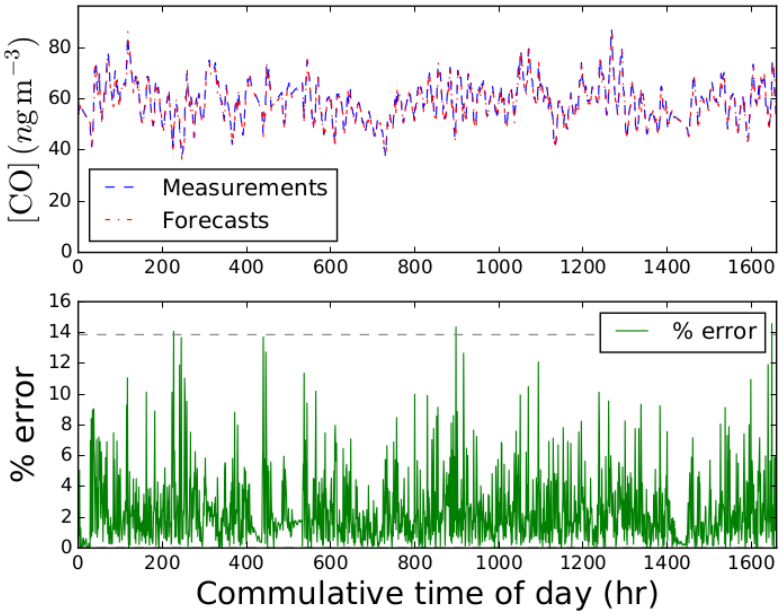
Table 4.2: Machine learning CO forecasting results. NOTE: all values are in percentages except for the standard deviation (Stdev).

CO					
			MAPE		
	Test 1	**Test 2**	**Test 3**	**Mean**	**Stdev**
Original	7.748	7.728	7.680	7.719	0.035
7 day filter	9.531	9.946	11.427	10.302	0.996
28 day filter	12.658	13.160	13.861	13.227	0.604
Savitzky-Golay	7.475	7.455	7.407	7.446	0.035

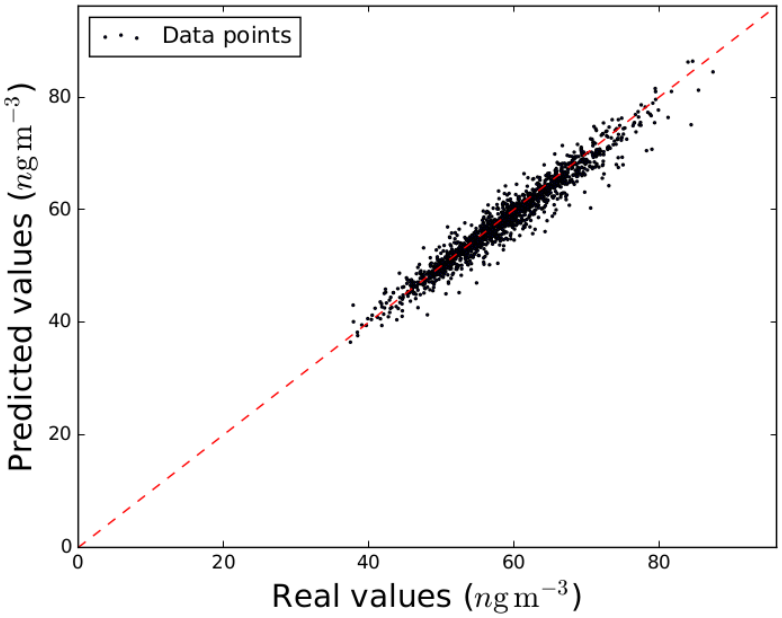
Overall, this model seems to have sacrificed accuracy in order to improve consistency from the previous model.

The final model tested for carbon monoxide forecasting, used a Savitzky-Golay filter on the original signal and managed to produce the following MAPE values of 7.475 %, 7.455 % and 7.407 % during testing. These values are slightly lower than those of the original model and significantly lower than the moving average based ones. The consistency of the forecasts is verified by the standard deviation of 0.035 and the analysis of the right-sided plot in figure 4.4 that shows most data points over the quadrant bisection indicating good correlation between measurements and predictions. Good model accuracy is indicated by the mean MAPE value of 7.446 % and by the fact that the highest individual error, shown in the left-sided plot of figure 4.4, did not reach that values.

In table 4.2 are presented the performance metrics relative to all four models trained for carbon monoxide forecasting. Both the first and last models had the same very low results regarding consistency of forecasts, as viewed by their standard deviations, indicating that their forecasting abilities do not depend on the input shape. The second and third models had higher standard deviations, which means that they are both less consistent, however the standard deviations are still relatively low so their forecasts are still quite consistent. In terms of accuracy both models based on the moving average filter had higher MAPEs indicating that they are less accurate than the original and polynomial filter models. Between the two, the model that used the Savitzky-Golay filter showed a slight improve in accuracy from the original model.

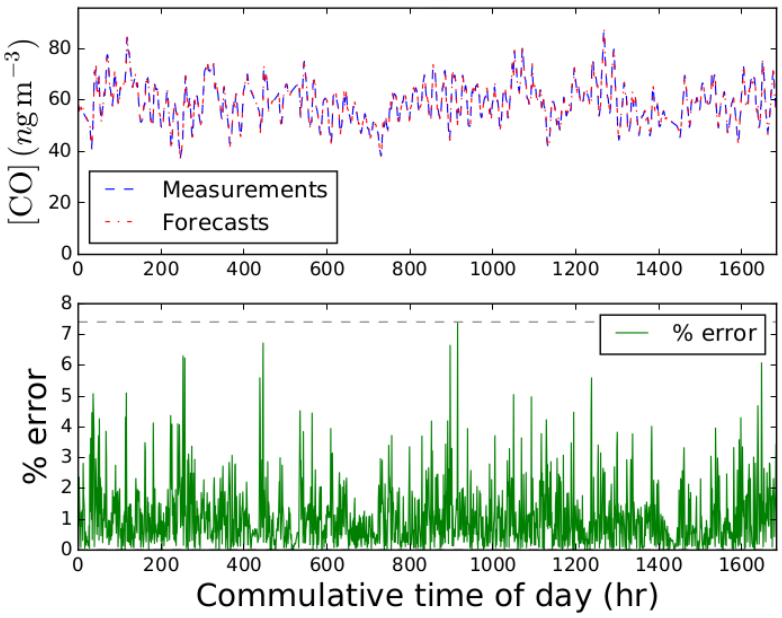


(a)

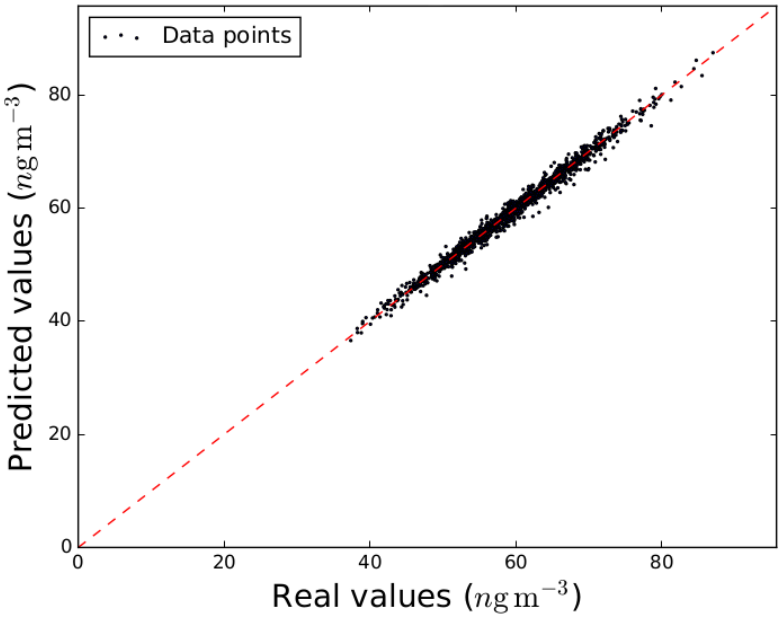


(b)

Figure 4.3: Smoothened 24 hour CO forecasting results for the third iteration using a moving average with a window of 28 *days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.*



(a)



(b)

Figure 4.4: Smoothened 24 hour CO forecasting results for the third iteration using 3^{rd} degree polynomial Savitzky-Golay filter. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

4.2 Nitrogen dioxide

The nitrogen dioxide data set was 95.628 % complete with only the remaining 4.37 % being comprised of absent measurements. The mean value of the measurements prior to data inference was of 0.000889 ngm^{-3} with a standard deviation of 0.000608 and were slightly increased to 0.000939 ngm^{-3} and 0.000640 respectively which shows that signal behavior remained relatively the same after inferring missing data points. The critical values used for the Dickey-Fuller test were:

- Critical Value (10%) = −2.566946
- Critical Value (5%) = −2.861870
- Critical Value (1%) = −3.431097

And the test confirmed that all four signals were stationary with a 99 % confidence level meaning they are viable time-series for model training. Detailed results are presented in table 4.3. On a first impression, the p-values obtained from testing all signals are very close to zero which indicates that the data sets contained sufficient information to draw a conclusion. However, the most important part is the fact that all test values are inferior to the critical values, hence the null hypothesis can be discarded confirming the necessary signal stationarity for further use.

4.2.1 NO_2 model performance

Training the model with the original signal produced the results shown in figure 4.5. Individual mean absolute percentage errors for each testing iteration were of 206.086 %, 189.127 % and 130.610 % that averaged at 175.274 % with a standard deviation of 39.599. These results are indicative of a poor model that fails to give

Table 4.3: Dickey-Fuller test results for NO_2 signals.

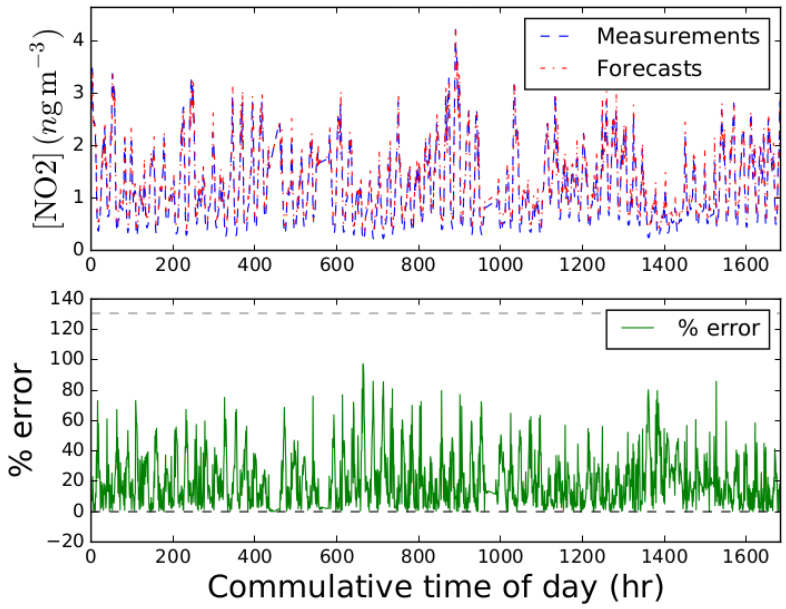
	Observations	Test value	P-value	Stationary
Original	8759	-11.42458	6.74E-21	Yes
7 day filter		-13.31858	6.53E-25	Yes
28 day filter		-12.32541	6.61E-23	Yes
Savitzky-Golay filter		-10.32594	2.93E-18	Yes

accurate or consistent predictions. Individual errors were very high and frequent, as shown by the left-sided plot, and although they did not the reach the MAPE value the fact they were consistently high contributed for the rise in that value. The scatter plot in the right side of the figure shows poor correlation between real and forecasted values since the core data points do not overlap with the quadrant bisection.

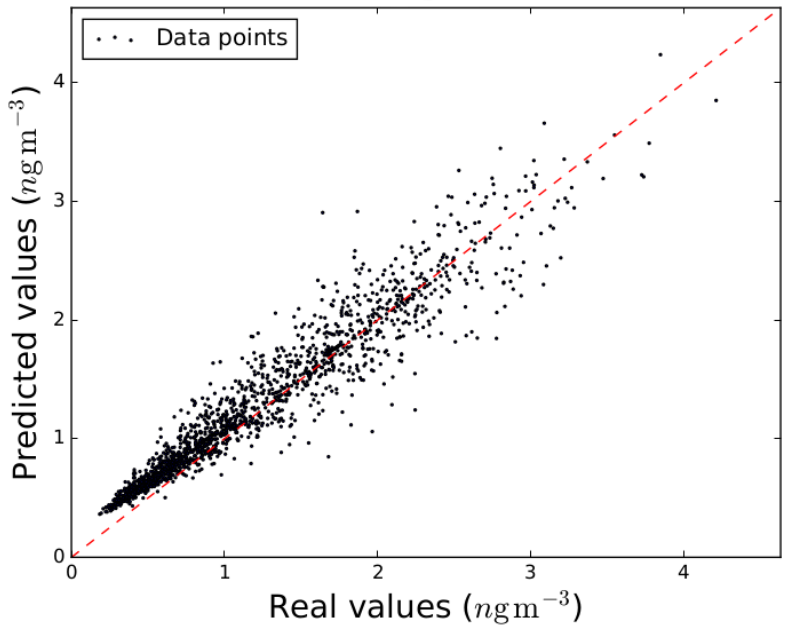
Using a moving average filter with a seven day window managed to decrease the testing MAPE values to 180.083 %, 163.994 % and 136.506 % with the mean value being of 160.195 % with a standard deviation of 22.035. On the other hand, using a 28 day window yielded individual MAPE values of 195.346 %, 178.858 % and 123.223 % making the mean value equal to 165.809 % with a standard deviation of 37.791. Although there is some improvement in both these situations, model performance and accuracy still remains unsatisfactory. The plots from the first filter are present in figure 4.6 where it is visible that no particular improvement has been made, although the frequency of high individual is lower and the correlation plot shows the core data points closer to the quadrant bisection. The ones in figure 4.7 are relative to the 28 day moving average filter and show even less improvement than the previous filter, with the frequency of individual errors being high and the core data points in the dispersion plot still failing to overlap the quadrant bisection.

Lastly, the Savitzky-Golay filter was used prior to algorithm training. This generated individual MAPE values of 201.177 %, 184.112 % and 126.937 % which seem similar to the original model. The average value is of 170.743 % with a standard deviation of 38.884 which seems to confirm this assumption. However, analysis of the graphics in figure 4.8 indicates that the frequency and size of individual errors has decreased and that coherence between forecasts and measurements has increased suggesting a model improvement.

Performance metric obtained from the model built to forecast nitrogen dioxide concentration are presented in table 4.4. Regarding model accuracy and consistency, none of the models presented seems viable for prediction NO_2 concentrations because MAPE values are very high and standard deviation values indicate that the models are very dependent on the shape of the input data series. Although, a graphical analysis indicates that the model built upon applying the Savitzky-Golay filter seems more efficient even if it tends to overestimate the real value in most situations.

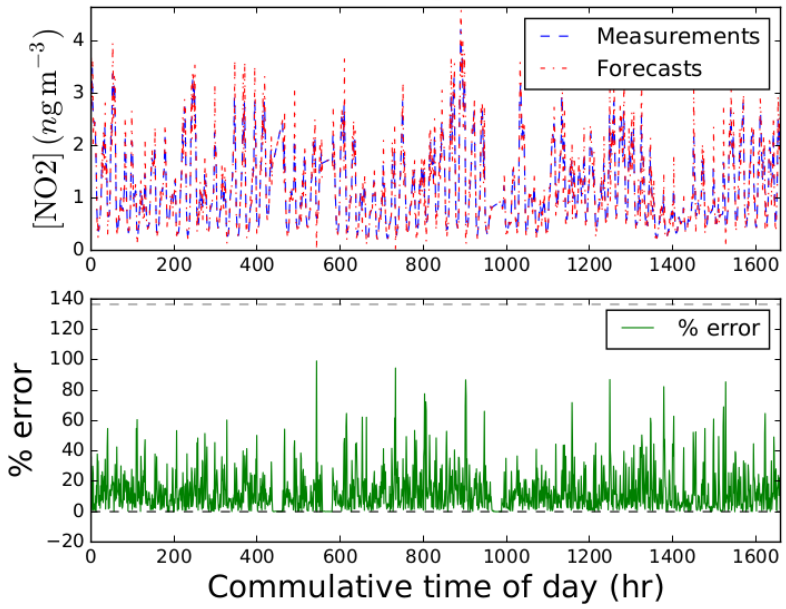


(a)

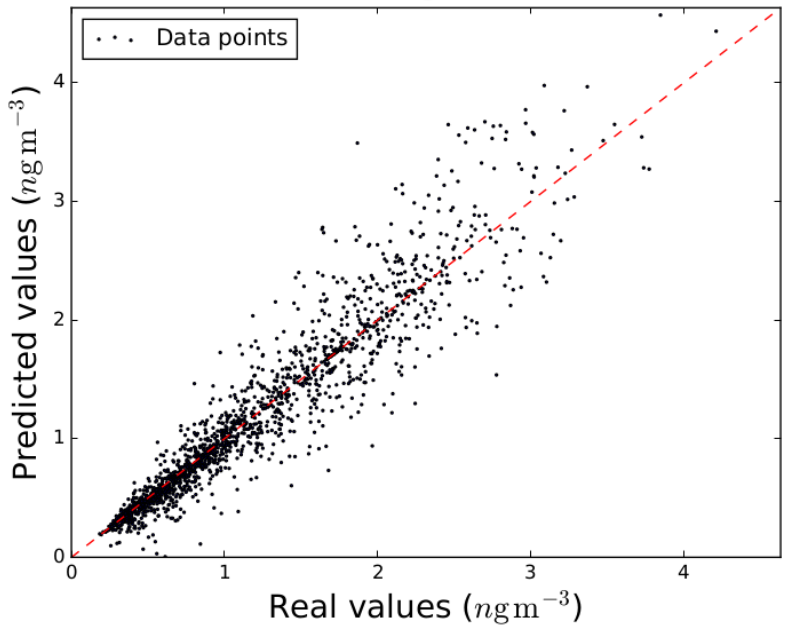


(b)

Figure 4.5: Un-smoothened 24 hour NO_2 forecasting results for the third iteration. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

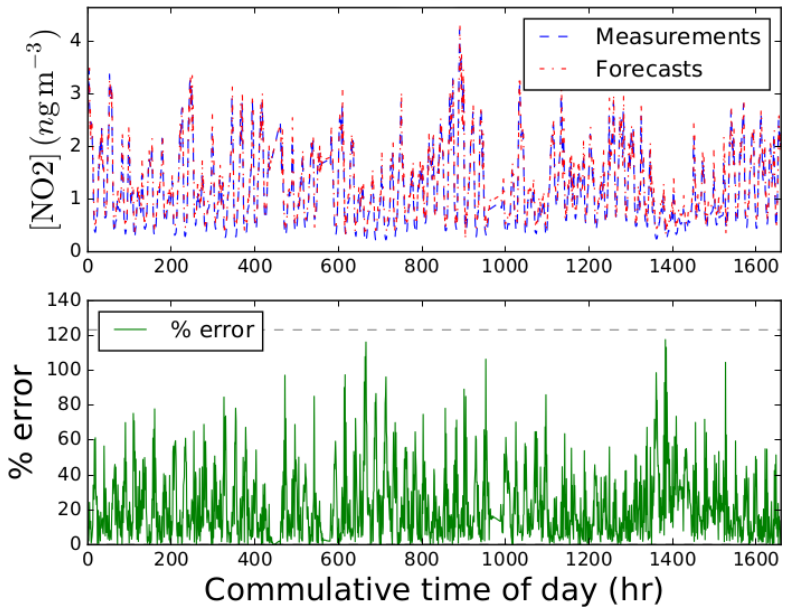


(a)

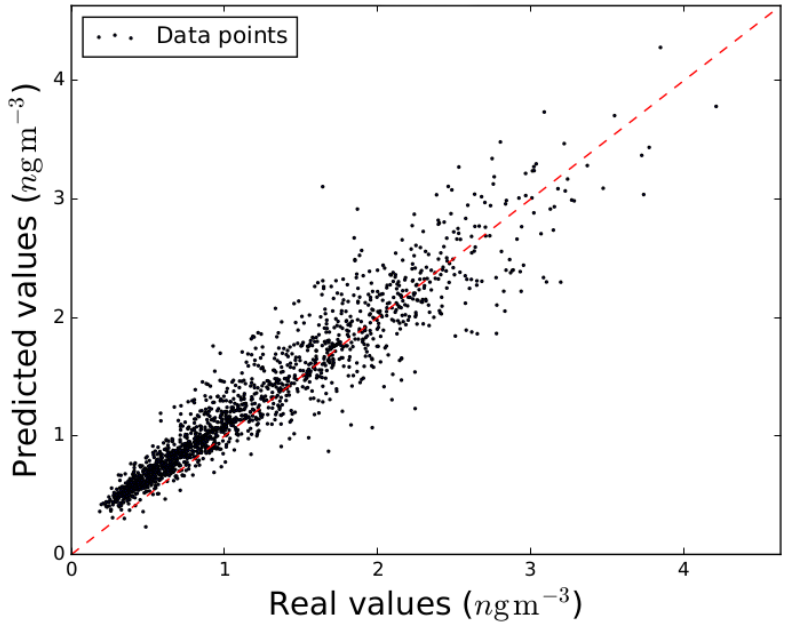


(b)

Figure 4.6: Smoothened 24 hour NO_2 forecasting results for the third iteration using a moving average with a window of seven days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

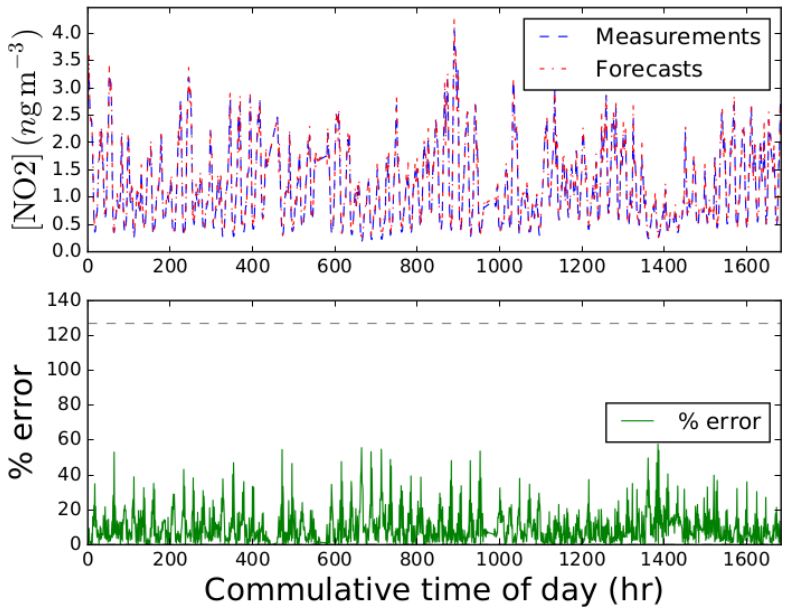


(a)

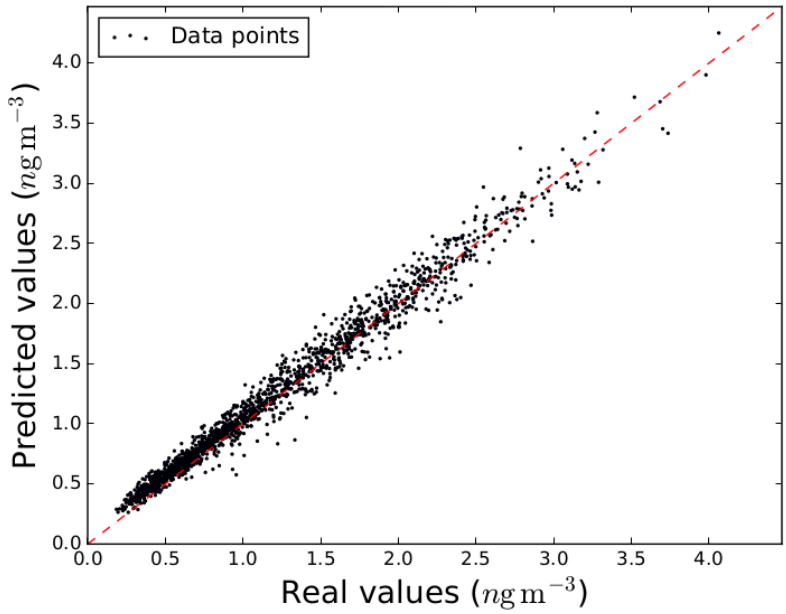


(b)

Figure 4.7: Smoothened 24 hour NO_2 forecasting results for the third iteration using a moving average with a window of 28 days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.



(a)



(b)

Figure 4.8: Smoothened 24 hour NO_2 forecasting results for the third iteration using 3^{rd} degree polynomial Savitzky-Golay filter. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

Table 4.4: Machine learning NO_2 forecasting results. NOTE: all values are in percentages except for the standard deviation (Stdev).

	NO_2				
	MAPE				
	Test 1	**Test 2**	**Test 3**	**Mean**	**Stdev**
Original	206.086	189.127	130.610	175.274	39.599
7 day filter	180.083	163.994	136.506	160.195	22.035
28 day filter	195.346	178.858	123.223	165.809	37.791
Savitzky-Golay	201.177	184.112	126.937	170.742	38.884

4.3 Tropospheric ozone

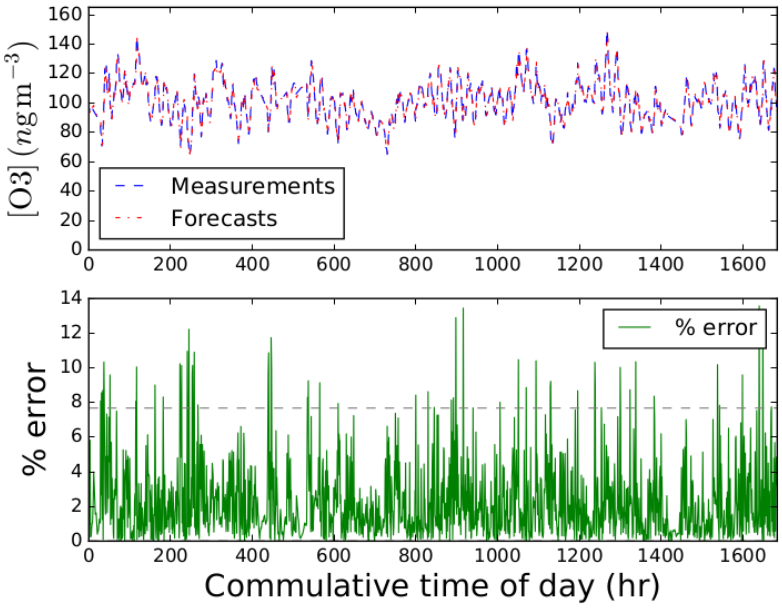
The tropospheric ozone signal had a completion of 93.112 % and its values averaged at 0.0394 ngm^{-3} with a standard deviation of 0.00415. After inferring the missing data points, these values changed to 0.0424 ngm^{-3} and 0.00446 respectively indicating that its behavior did no changed significantly. Concerning stationarity, the critical values used were:

- Critical Value (10%) = −2.566946
- Critical Value (5%) = −2.861870
- Critical Value (1%) = −3.431097

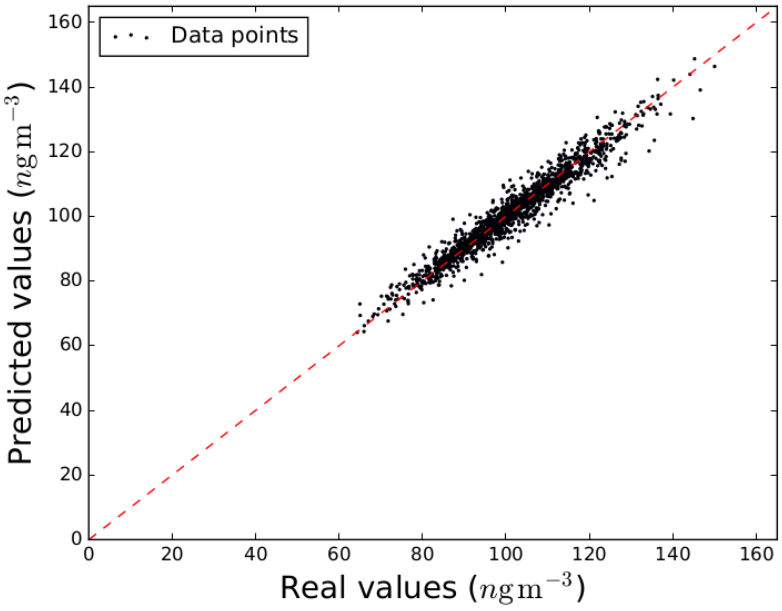
And the Dickey-Fuller test results indicated that all four signals were stationary with a 99 % confidence level. Table 4.5 shows these results and it is visible that p-values obtained from every signal are small and close to zero which means that the data sets are informative enough for the conclusion of the test to hold under any circumstances. With that being established, all test values and inferior to the 1 % critical value confirming the stationarity of the time-series.

4.3.1 O_3 model performance

The plots presented in figure 4.9 indicate how the model behaved when forecasting tropospheric ozone concentrations using the original time-series in the training phase. Individual MAPE values for each testing iteration were 7.748 %, 7.728 % and 7.680 % which are fairly small indicating a good model. The left-sided plot shows



(a)



(b)

Figure 4.9: Un-smoothened 24 hour O_3 forecasting results for the third iteration. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

Table 4.5: Dickey-Fuller test results for O_3 signals.

	Observations	Test value	P-value	Stationary
Original	8746	-5.148123	1.10E-05	Yes
7 day filter		-12.91134	4.20E-05	Yes
28 day filter		-9.216728	4.05E-24	Yes
Savitzky-Golay filter		-4.859107	1.83E-15	Yes

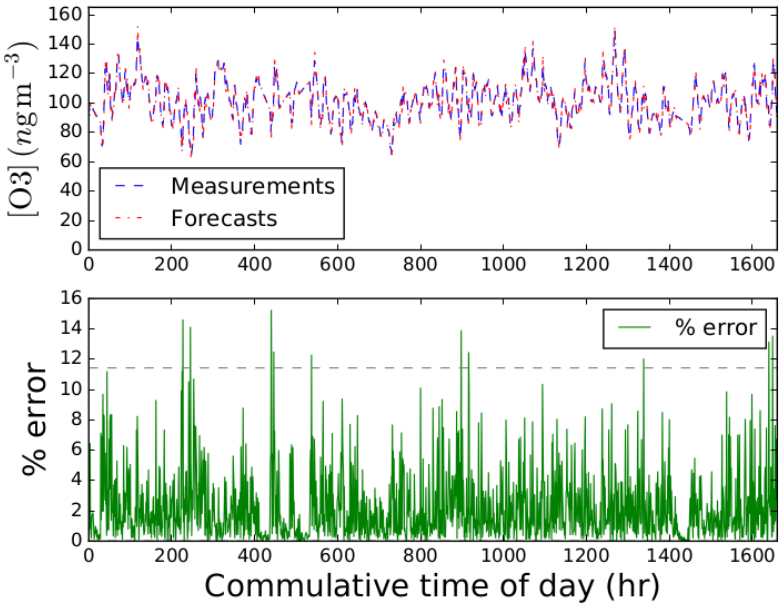
good similarity between forecasts and real values and that individual errors did no exceed 14 % while the right-sided plot further supports that coherence between forecasts and real values is indeed present. Model performance is evaluated based on the mean MAPE value of 7.719 % with the standard deviation being equal to 0.035.

Applying the moving averages filter, first with the seven day window and second with the 28 day window, produced models whose results are respectively presented in figures 4.10 and 4.11a respectively.

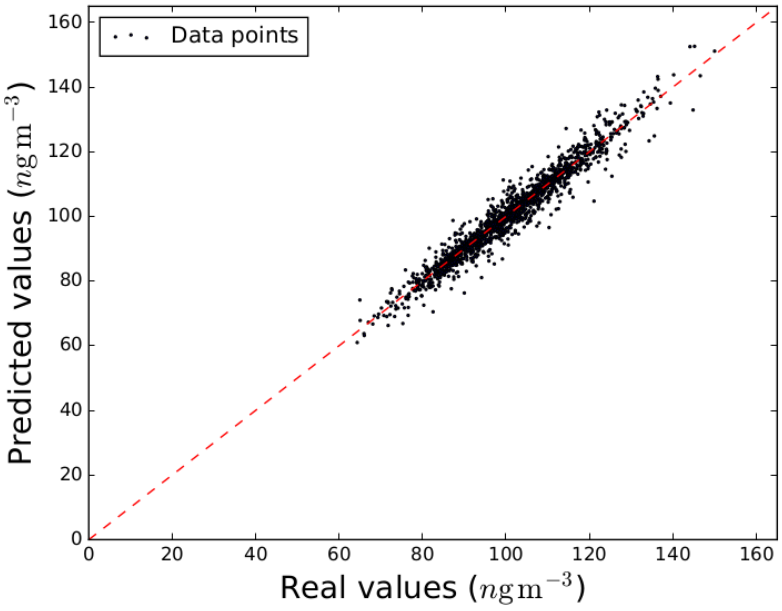
Looking first at the results from the application of the first window, individual MAPE values were of 9.531 %, 9.946 % and 11.427 % who averaged at 10.302 % with a standard deviation of 0.996. These values indicate that both accuracy and consistency measures for the forecasted values diminished relatively to the original model. However, looking at the pots themselves, individual errors seem not much higher and correlation between forecasted and real values is still adequate.

The application of the second window, had similar effects since it increased individual MAPE values obtained during testing to 12.658 %, 13.160 % and 13.861 % placing model performance at the mean MAPE value of 13.227 % with a standard deviation of 0.604. These values imply that there was no improvement from using these filter, however with this window size forecasts, while less accurate were more consistent than with the previous window size. Plot analysis seems to corroborate these assumptions showing that individual errors retained their values and frequencies and that coherence between forecasts and measurements is consistent.

A fifth polynomial Savitzky-Golay filter managed to reduce the original individual MAPE values to 7.475 %, 7.455 % and 7.407 %. Its average value is of 7.446 % with the respective standard deviation of 0.035 revealing a slight improvement on the original model in terms of accuracy and forecasting consistency. Its plots are shown in figure 4.12 and it is possible to see that there was a significant reduction on both



(a)



(b)

Figure 4.10: Smoothened 24 hour O_3 forecasting results for the third iteration using a moving average with a window of seven days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

Table 4.6: Machine learning O_3 forecasting results. NOTE: all values are in percentages except for the standard deviation (Stdev).

	O_3				
	MAPE				
	Test 1	**Test 2**	**Test 3**	**Mean**	**Stdev**
Original	7.748	7.728	7.680	7.719	0.035
7 day filter	9.531	9.946	11.427	10.302	0.996
28 day filter	12.658	13.160	13.861	13.227	0.604
Savitzky-Golay	7.475	7.455	7.407	7.446	0.035

the size and frequency of the individual errors for each forecast. Also, the correlation plot between real and predicted values places most data points over the quadrant bisection meaning that model consistency was achieved.

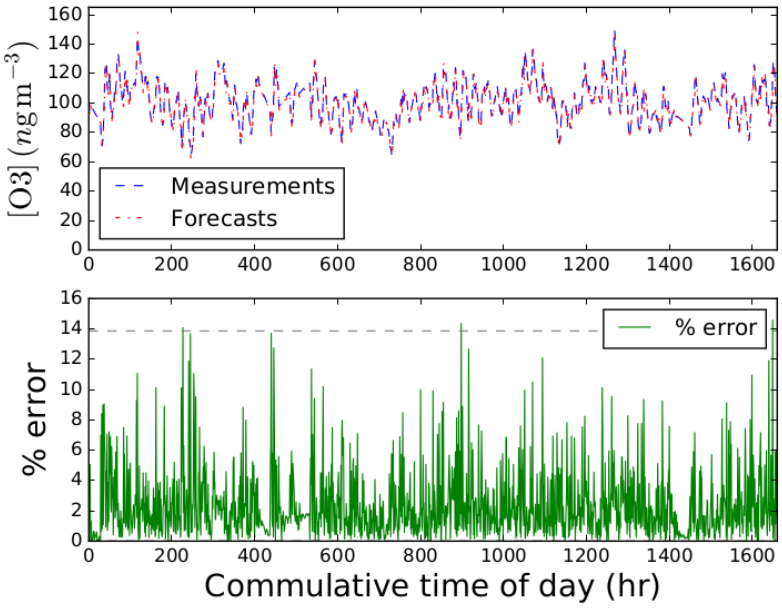
Table 4.6 is a condensed form of all the results discussed for far in this section. It is noticeable that both filters that used a moving averaged did not manage to improve upon the original signal for constructing the machine learning algorithm for tropospheric ozone forecasting. However, both these models still have adequate accuracy and consistency. As for the Savitzky-Golay filter it managed to outperform the original model in terms of accuracy while retaining the same level of consistency because of its smaller MAPE and equal standard deviation.

4.4 Sulfur dioxide

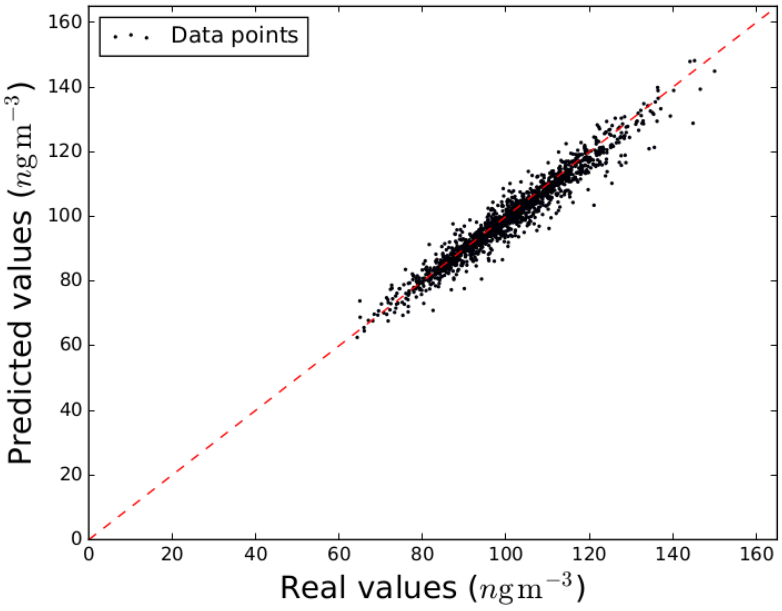
About 9.0 % of the data regarding the sulfur dioxide signal was missing and prior to data inference its mean value was of $0.00193\,\mathrm{ngm}^{-3}$ with the respective standard deviation of 0.000307. After inferring the missing data points these values remained consistent at $0.00196\,\mathrm{ngm}^{-3}$ and 0.000339 respectively meaning that signal behavior did not change significantly. Which implies that the signal is good enough for training a machine learning model.

The complete signal was submitted for stationarity verification via the Dickey-Fuller method, using the following critical values:

- Critical Value (10%) = -2.566946
- Critical Value (5%) = -2.861870

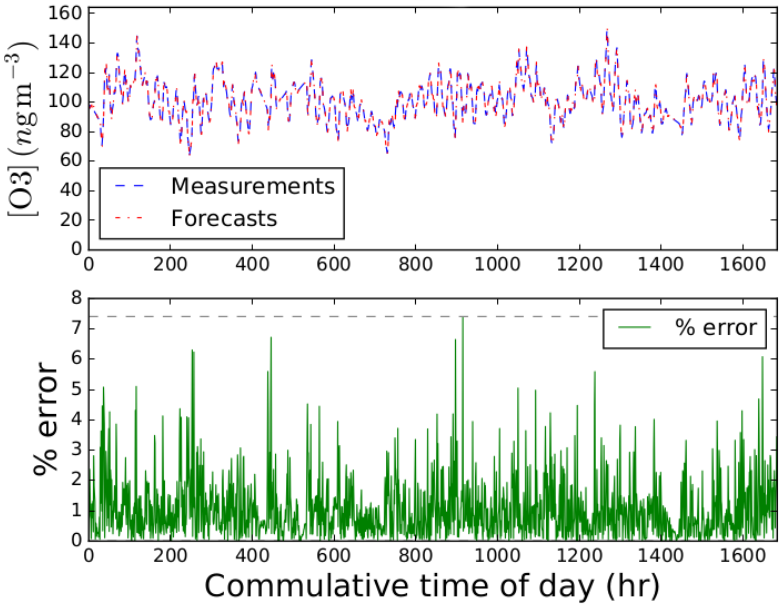


(a)

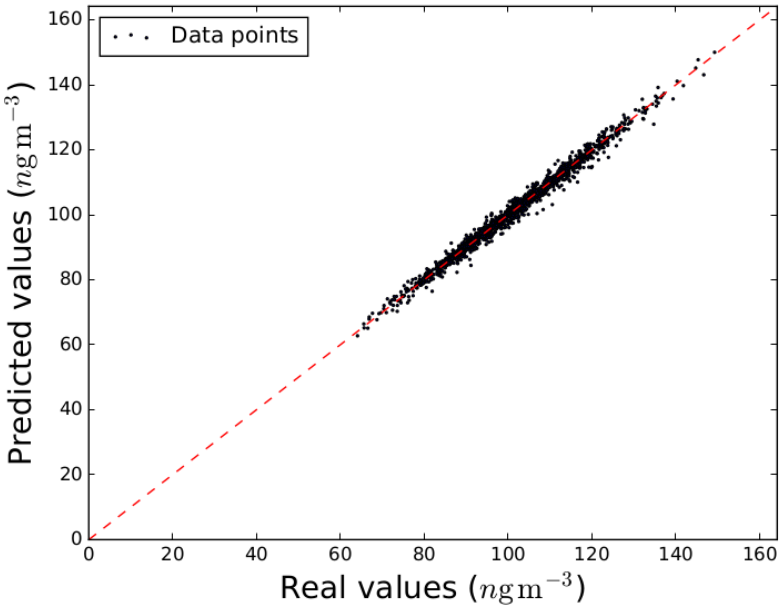


(b)

Figure 4.11: Smoothened 24 hour O_3 forecasting results for the third iteration using a moving average with a window of 28 days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.



(a)



(b)

Figure 4.12: Smoothened 24 hour O_3 forecasting results for the third iteration using 3^{rd} degree polynomial Savitzky-Golay filter. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

Table 4.7: Dickey-Fuller test results for SO_2 signals.

	Observations	Test value	P-value	Stationary
Original	8783	-93.7159	6.17E-25	Yes
7 day filter		-13.3186	6.53E-25	Yes
28 day filter		-14.0261	3.50E-26	Yes
Savitzky-Golay filter		-14.7039	2.92E-27	Yes

- Critical Value (1%) = -3.431097

which confirmed that all sulfur dioxide data sets were stationary with a 99 % confidence level. Results are presented in table 4.7 and indicate very low p-values for every data set testing indicating that they contained enough information about the system to make the test significant. Also, since all test values are inferior to the 1 % critical value, signal stationary is confirmed for all cases.

4.4.1 SO_2 model performance

Individual MAPE values obtained during the testing phase of the model using the original signal were of 42.9015 %, 43.202 % and 43.194 %. This last iteration is presented graphically by the plots in figure 4.13 where it is possible to notice that there occurred a few sporadic very high individual errors while forecasting sulfur dioxide values, one of which that surpassed the 160 % mark. The correlation plot resembles a disperse blob and while the core points overlap the quadrant bisection there are still many data points outside of it. These factors indicate a possible viable model, however also very unstable regarding its predictions. The fact that there is a larger absence of data than with the previous compounds may also have affected model performance: a MAPE value of 43.099 % with a standard deviation of 0.1712.

The application of the moving average based filters yielded the results presented in figures 4.14 and 4.15a for the seven and 28 day windows respectively. Given how much they differ a separate analysis is necessary.

The seven moving average filter, MAPE values retrieved during testing were 180.083 %, 163.994 % and 136.506 % which places model performance at a mean MAPE value of 160.195 % with the standard deviation of 22.035. These results indicate a poor a unusable model. In a similar way, the model built upon the 28 day moving average filter yielded individual MAPE values of 882.898 %, 886.269 % and

Table 4.8: Machine learning SO_2 forecasting results. NOTE: all values are in percentages except for the standard deviation (Stdev).

	SO_2 MAPE				
	Test 1	**Test 2**	**Test 3**	**Mean**	**Stdev**
Original	42.901	43.202	43.194	43.099	0.171
7 day filter	180.083	163.994	136.506	160.195	22.035
28 day filter	882.898	886.269	884.720	884.629	1.688
Savitzky-Golay	41.272	41.573	41.567	41.471	0.172

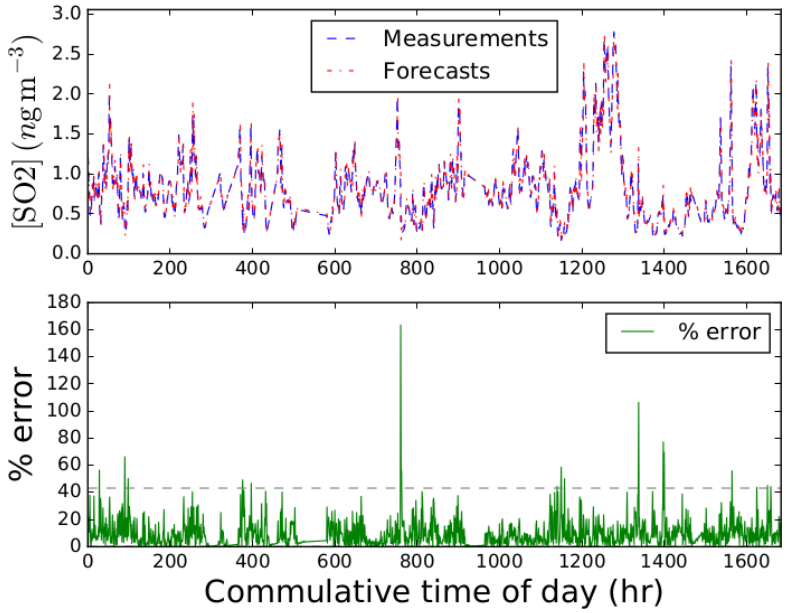
884.720 % which are much higher than even the previous values. The mean value is used to describe this model performance at 884.629 % with the respective standard deviation of 1.688, meaning that this model is even less usable than the previous one.

Using the Savitzky-Golay filter while creating the model yielded individual MAPE values of 41.273 %, 41.573 % and 41.567 % which are more similar to the values generated by the original model. By looking ate the plots in figure 4.16 it is noticeable that the model performed well and improved upon the original by lowering the frequency and size of the individual errors and increasing the consistency of the forecasts as shown by the correlation plot. The mean MAPE of 41.471 % with the standard deviation of 0.172 translates model performance.

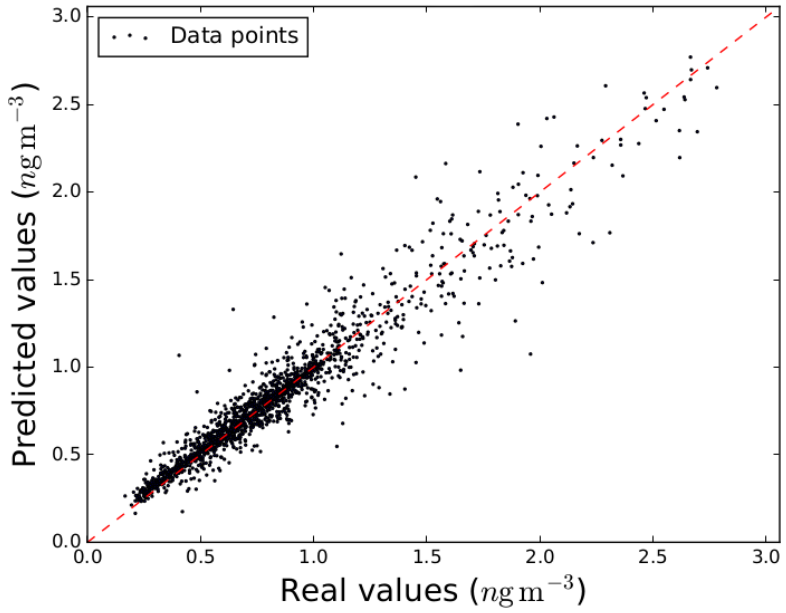
Looking at the values in summary form presented in table 4.8 it is possible to realize that both models based on the moving average filters for the prediction of sulfur dioxide concentrations are unusable given their high performance metrics. however, both the original model and the one based on the Savitzky-Golay filter appear to be usable despite their MAPE values with the later being slightly better in therms of accuracy and consistency given its better MAPE and standard deviation values.

4.5 Particulate matter 10 (PM10)

The data set regarding PM10 concentrations was 92.77 % complete. Initial mean and standard deviation values are of 14.7 ngm^{-3} and 3.57 respectively and after data inference they became 15.8 ngm^{-3} and 3.85 also respectively. The change in both

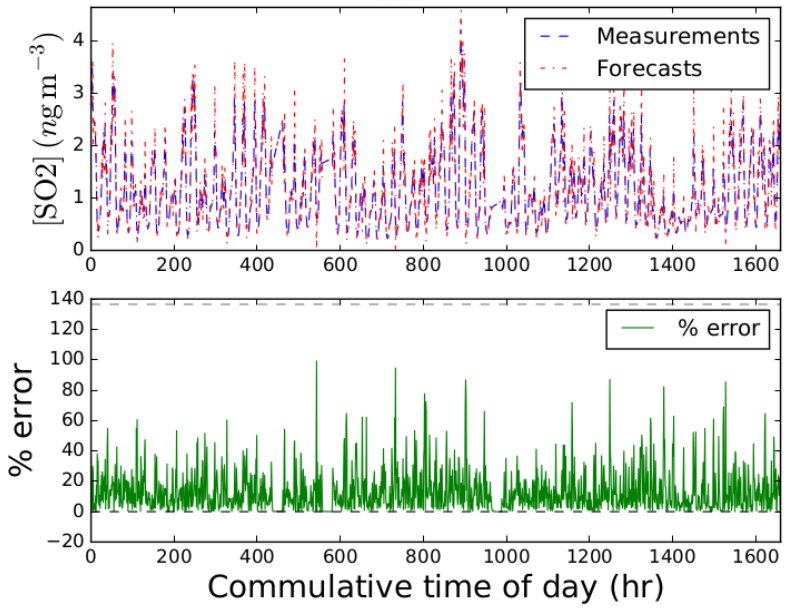


(a)

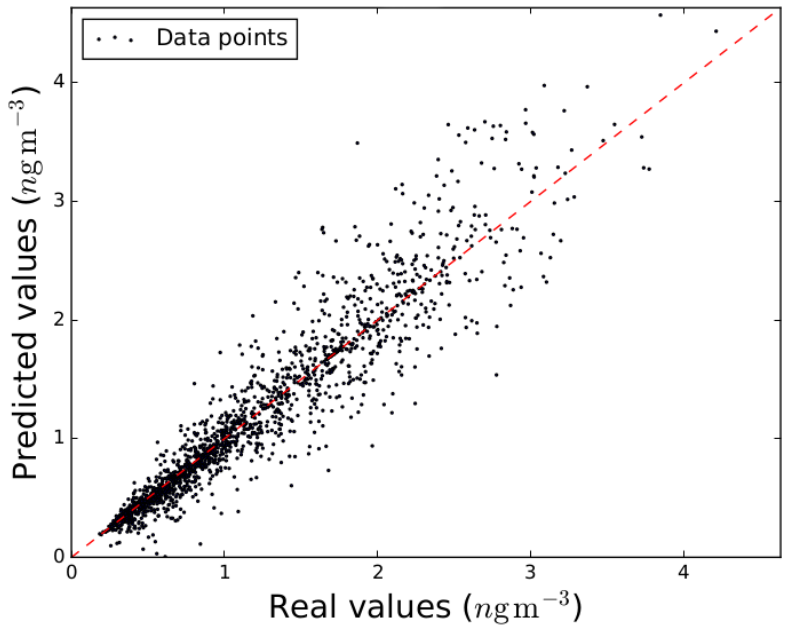


(b)

Figure 4.13: Un-smoothened 24 hour SO_2 forecasting results for the third iteration. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

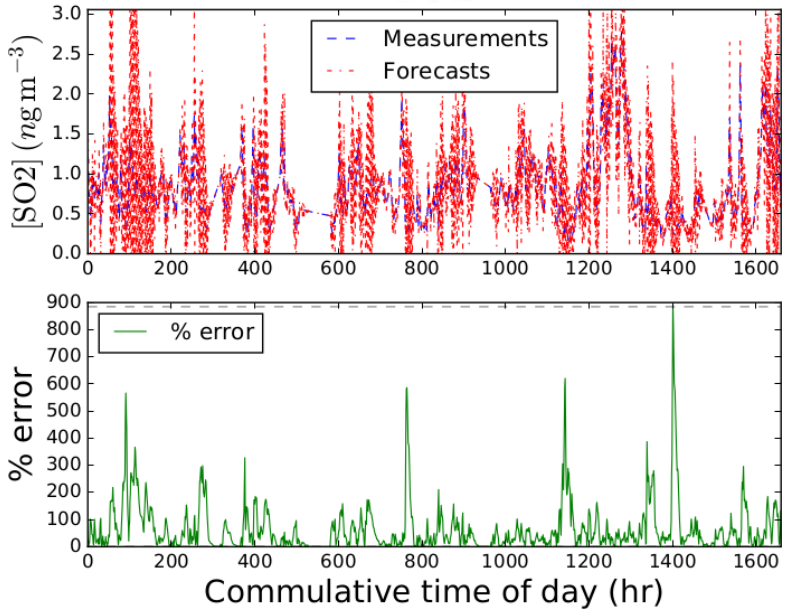


(a)

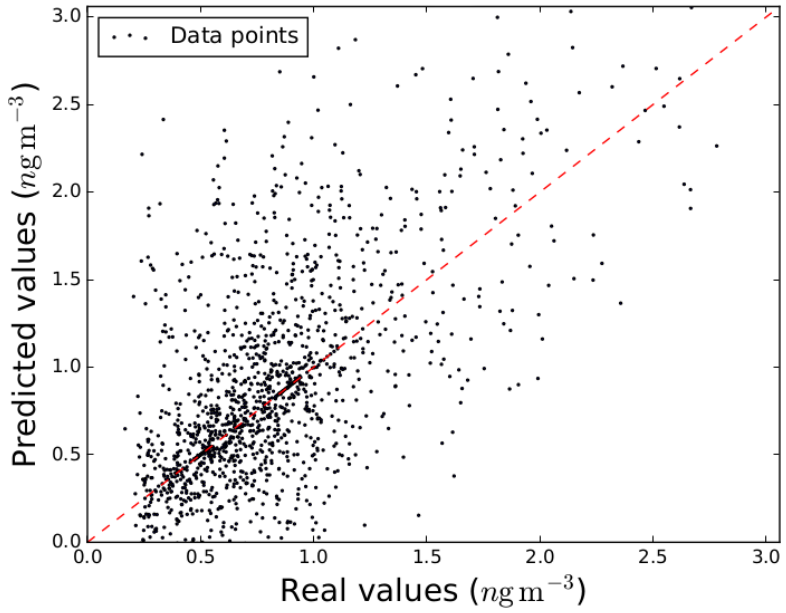


(b)

Figure 4.14: Smoothened 24 hour SO_2 forecasting results for the third iteration using a moving average with a window of seven days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

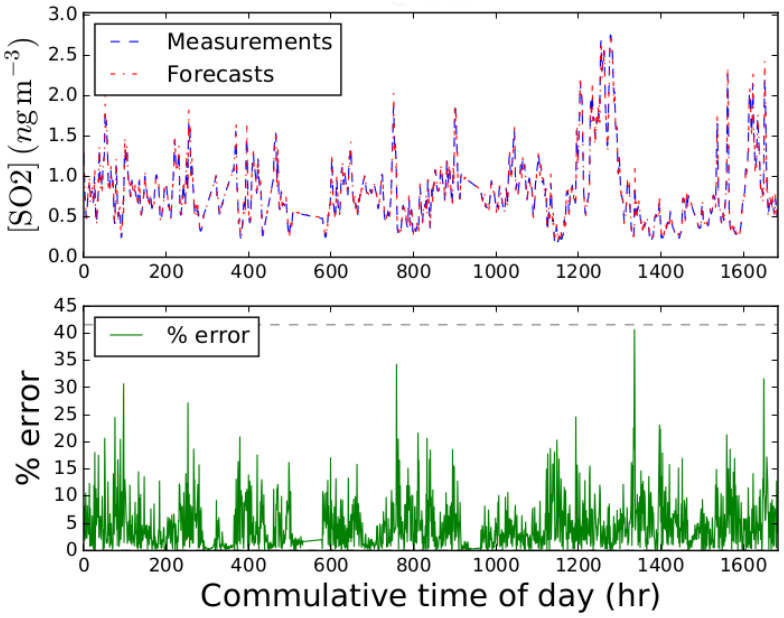


(a)

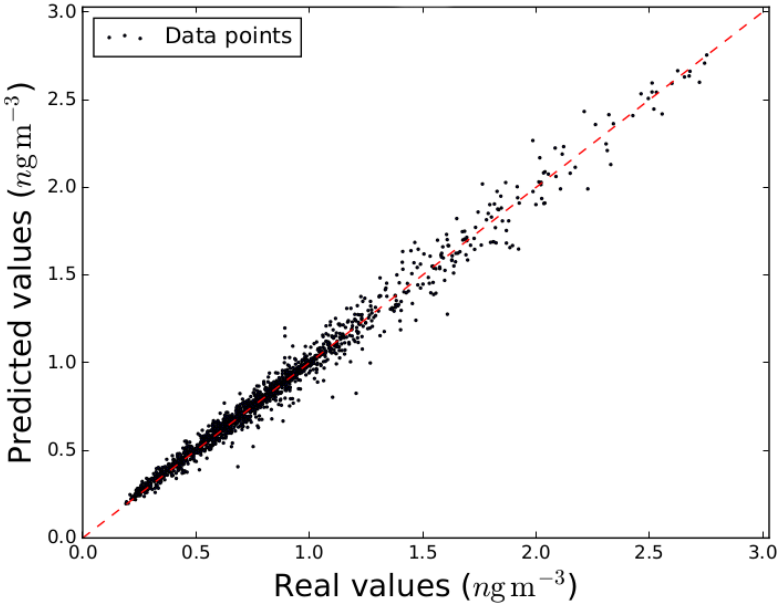


(b)

Figure 4.15: Smoothened 24 hour SO_2 forecasting results for the third iteration using a moving average with a window of 28 days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.



(a)



(b)

Figure 4.16: Smoothened 24 hour SO_2 forecasting results for the third iteration using 3^{rd} degree polynomial Savitzky-Golay filter. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

values is small enough to consider that signal behavior remains unchanged after inferring the missing data, which meant that it is viable for algorithm construction.

The complete data set was comprised of 8760 observations and the critical values used in the Dickey-Fuller test were:

- Critical Value (10%) = -2.566946
- Critical Value (5%) = -2.861870
- Critical Value (1%) = -3.431097

Table 4.9 shows the results of the Dickey-Fuller test on all signals. For the original signal the test value was of -9.38595, for the seven days moving average of -11.48578, for the 28 days moving average of -9.854643, and for the Savitzky-Golay filter of -8.291375. All these values are inferior to the 1 % critical value meaning that all signals are stationary with a 99 % confidence level. In addition, the fact that the p-values for all four signals are very small indicates that the sample population used provides enough evidence to reject the null hypothesis in its entirety. Hence, all signals can be used for machine learning model training, testing and evaluation.

4.5.1 PM10 model performance

Considering the model based on the original signal, the mean squared absolute percentage errors for the three testing iterations was of 27.667 %, 27.670 % and 27.281 % with the standard deviation of 0.224, which means that this model is coherent prediction-wise. The graphics in figure 4.17 pertain the third testing iteration for this model. An analysis of the left-sized plot shows that the high individual errors were few and scarce, with the highest individual error just surpassing the 80 %

Table 4.9: Dickey-Fuller test results for PM10 signals.

	Observations	Test value	P-value	Stationary
Original	8760	-9.38595	6.76E-16	Yes
7 day filter		-11.48578	4.87E-21	Yes
28 day filter		-9.854643	4.39E-17	Yes
Savitzky-Golay filter		-8.291375	4.25E-13	Yes

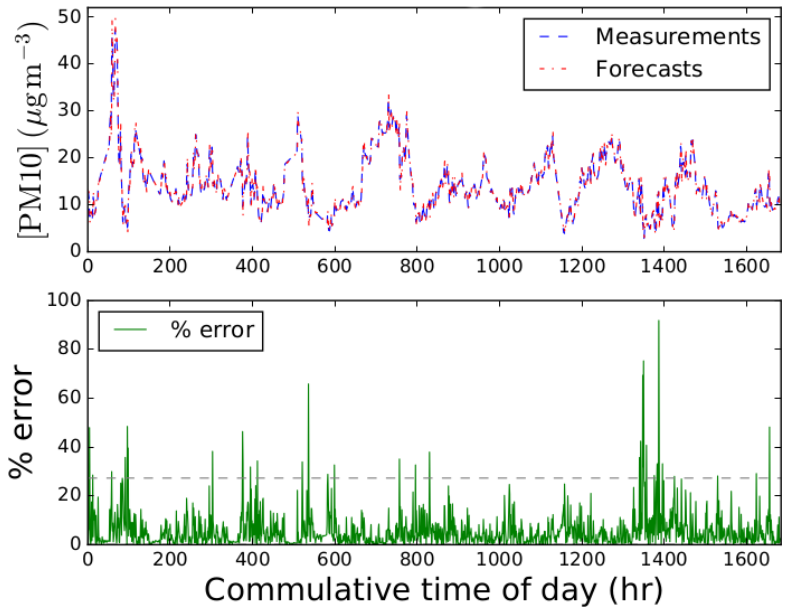
mark. The right-sized dispersion plot indicates that good correlation between the real and the forecasted values is present. Model performance is evaluated by the mean MAPE value of 27.539 %.

The model based on the seven day moving average resulted in very high mean absolute percentage errors: 504.841 %, 488.351 % and 498.171 %. However, its standard deviation, considering how large the mean MAPE was (497.121 %) is relatively small, 8.295, which means that despite lacking accuracy the model still retains coherence. The graphics in figure 4.18, further support this assumption since the individual errors shown are very high and the dispersion plot resembles a blob rather than overlapping the quadrant bisection.

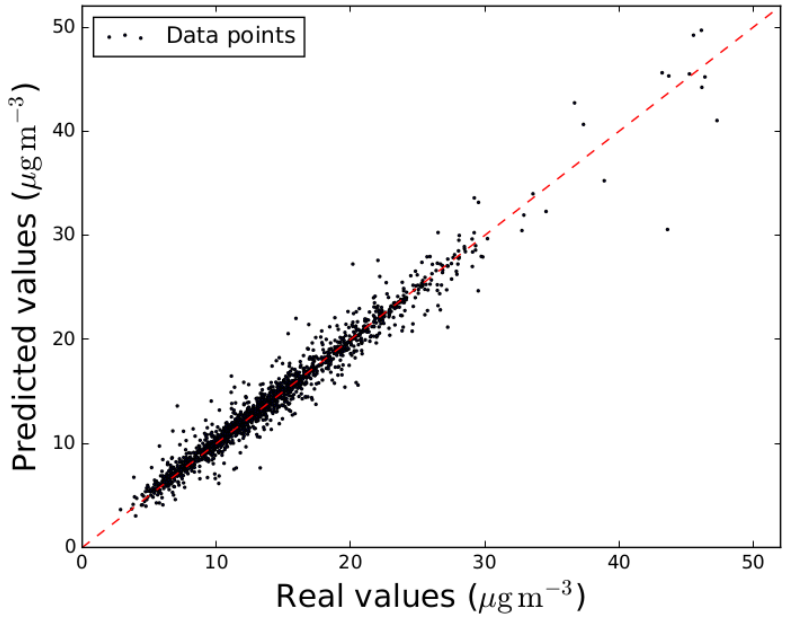
Using a 28 day moving average filter the individual MAPE values were of 27.315 %, 26.982 % and 27.974 %. Considering the standard deviation of 0.195, this model shows coherent forecasts similarly to the first model. A graphical analysis of the plots in figure 4.19 demonstrates a good correlation between real and forecasted values and lower individual errors than any other model so far, with the highest one not reaching the 80 % mark. The mean MAPE value used for model evaluation is of 27.09 %, slightly higher than the first model. Overall, this particular model improves both the coherence and the accuracy displayed by the original even if only slightly.

lastly, the signal smoothened using a Savitzky-Golay filter was tested and gave the mean absolute percentage errors of 26.563 %, 26.591 % and 26.200 %. On a first look these are the best individual MAPE values given so far. The standard deviation is of 0.218 which means that the model performs coherently. Analysis of the plots in figure 4.20 shows that high individual errors are scarce with the highest barely surpassing 25 % which is a significant improvement. Also, there exists a higher correlation between real and forecasted. The mean MAPE of 26.452 % is the lowest so far.

Table 4.10 suggests that the model using a seven day moving average filter is inadequate for using in a real scenario since it produced the worse results. The models that used the original signal and the signal decomposed using a 28 days moving average filter generated similar results, with the latter being slightly better in terms of accuracy and coherence. However, the model built on the Savitzky-Golay filter produced the best results in terms of accuracy, despite having slightly worse coherence than the model that used the 28 day moving average filter.

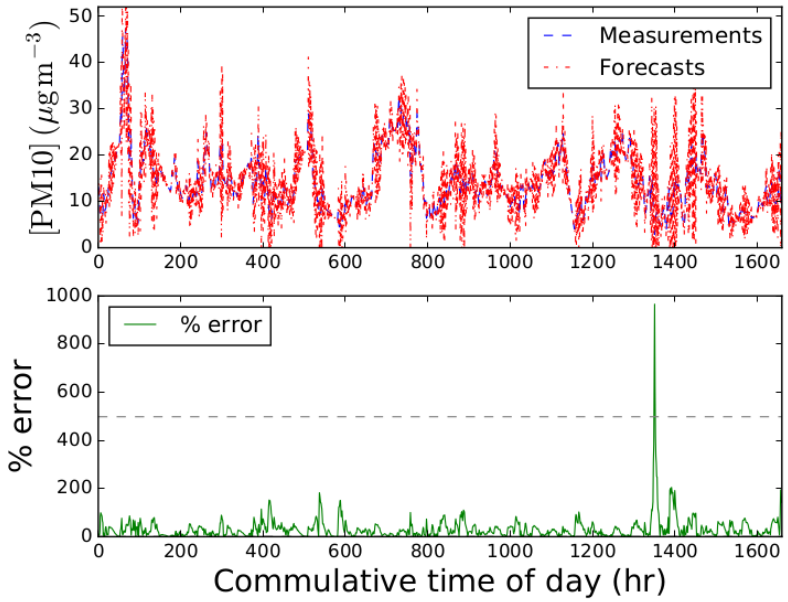


(a)

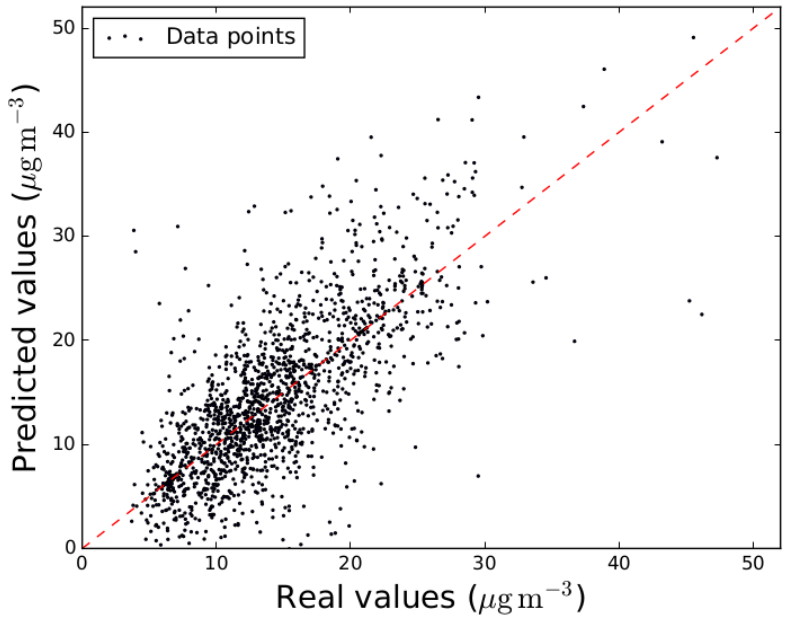


(b)

Figure 4.17: Un-smoothened 24 hour PM10 forecasting results for the third iteration. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

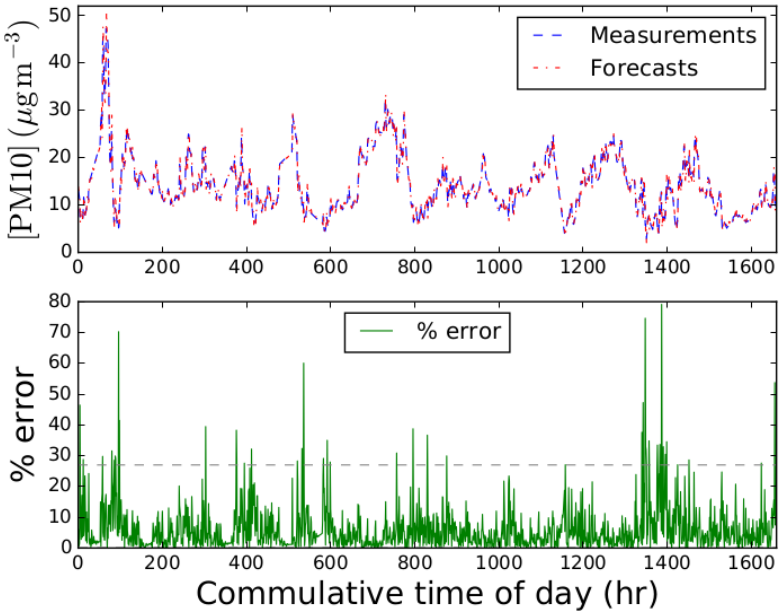


(a)

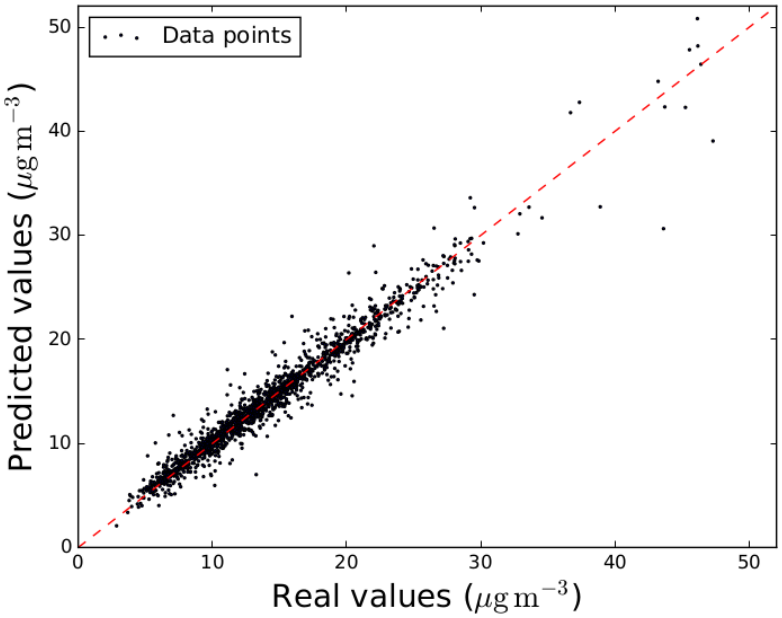


(b)

Figure 4.18: Smoothened 24 hour PM10 forecasting results for the third iteration using a moving average with a window of seven days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

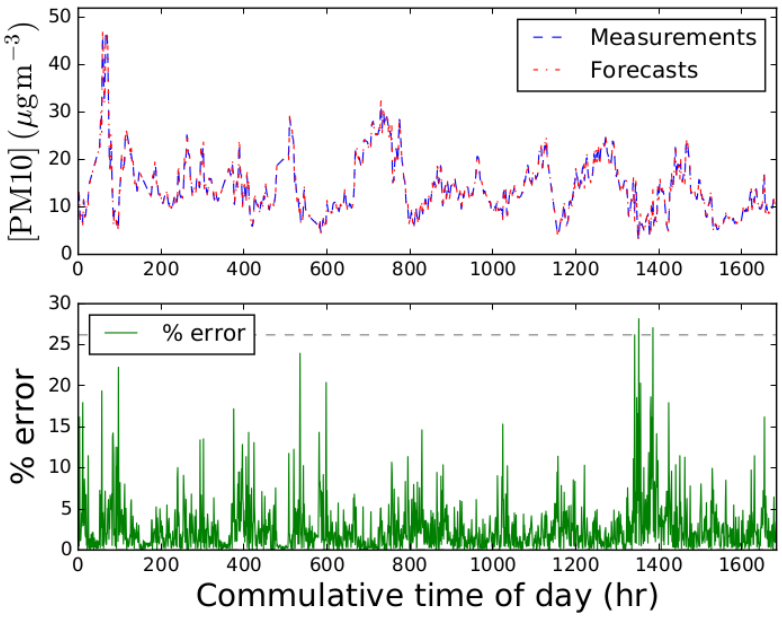


(a)

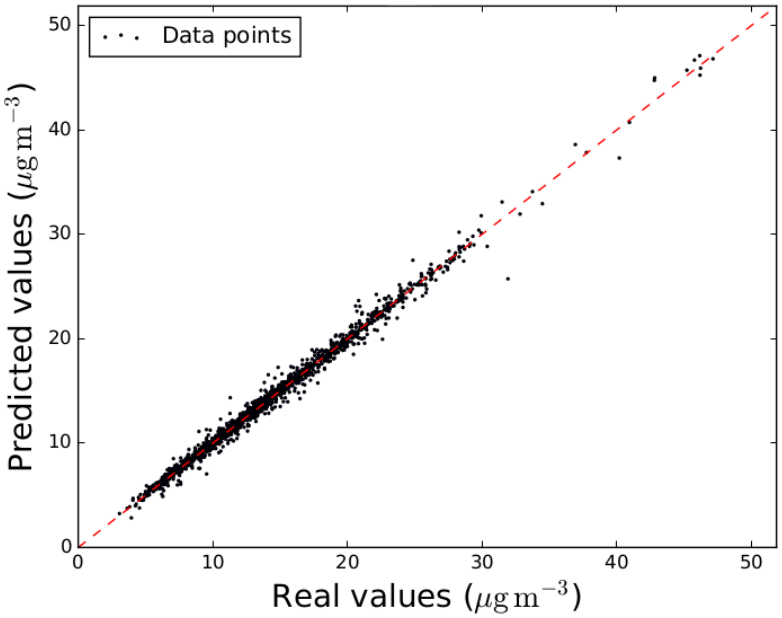


(b)

Figure 4.19: Smoothened 24 hour PM10 forecasting results for the third iteration using a moving average with a window of 28 *days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.*



(a)



(b)

Figure 4.20: Smoothened 24 hour PM10 forecasting results for the third iteration using 3^{rd} degree polynomial Savitzky-Golay filter. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

4.6 Particulate matter 2.5 (PM2.5)

The data set containing PM2.5 measurements has a total of 8774 and only 6.83 % of the data was absent. Prior to data inference the mean concentration was of 7.41 ngm^{-3} with a standard deviation of 2.54 that afterwards became 8.00 ngm^{-3} and 2.74 respectively. This indicates that signal behavior did not change greatly after inferring the missing data. So, the complete data set is suitable for trainig a MLA.

To verify if the data set was stationary, the critical values used in the Dickey-Fuller test are:

- Critical Value (10%) = −2.566945
- Critical Value (5%) = −2.861869
- Critical Value (1%) = −3.431096

And its results are presented in table 4.11. The p-values for every test are very close to zero indicating that the sample population was sufficient to reject or not the null hypothesis. For the original, seven day moving, 28 day moving averages and Savitzky-Golay filtered data sets the test values were of −10.69194, −11.35856, −10.62376 and −9.05221 respectively. Since all these values are bellow the 1 % critical value all series are stationary with a 99 % confidence interval.

Table 4.10: Machine learning PM10 forecasting results. NOTE: all values are in percentages except for the standard deviation (Stdev).

	PM10				
			MAPE		
	Test 1	**Test 2**	**Test 3**	**Mean**	**Stdev**
Original	27.667	27.670	27.281	27.539	0.224
7 day filter	504.841	488.351	498.171	497.121	8.295
28 day filter	27.315	26.982	26.974	27.090	0.195
Savitzky-Golay	26.563	26.591	26.200	26.451	0.218

4.6.1 PM2.5 model performance

The model built using the original signal had a mean absolute percentage errors of 30.880 %, 30.649 % and 30.369 % resulting in a mean MAPE value of 30.632 % with a standard deviation of 0.256. These values indicate adequate forecasting accuracy and a coherent behavior regarding its input. Graphical analysis of the results, presented in figure 4.21, shows that there is good accuracy, demonstrated by the existence of many small individual errors with few high errors. Also, the dispersion plot demonstrates that correlation between forecasted and real values is high given that most data points tend to overlap the quadrant bisection.

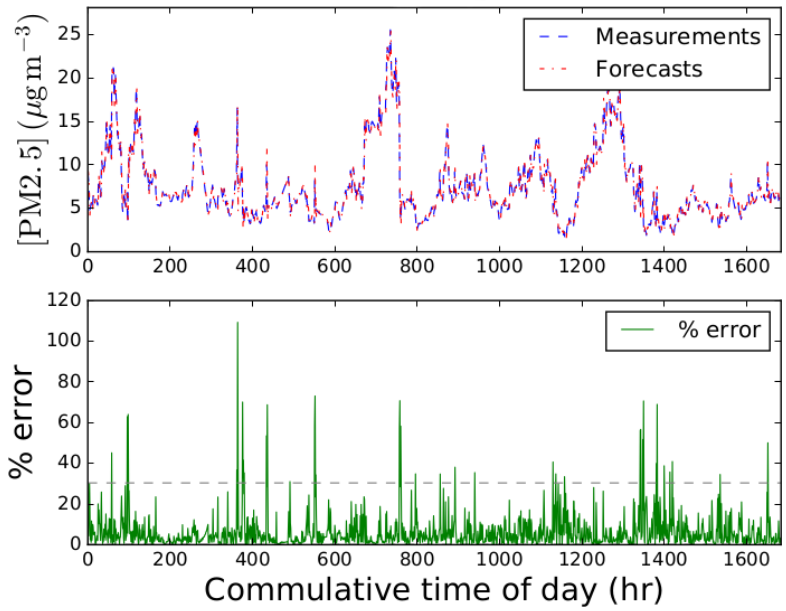
By using a seven day moving average filter on the data set prior to model training. the mean absolute percentage errors rose to 625.096 %, 612.060 % and 614.568 % resulting in an average MAPE of 617.241 % with a standard deviation of 6.917 %. These values are indicative of a very poor model that lacks accuracy while, however, still retaining a decent coherence. The plots in figure 4.22 further indicate that the model works poorly on forecasting PM2.5 values as indicated by the very high individual errors and the spread of the data points in the dispersion plot.

The model built using a 28 day moving average filter yielded results on par with those of the first model presented for PM2.5 forecasting. The mean absolute percentage errors were 32.764 %, 31.552 % and 31.066 % with its mean value being of 31.794 % and the corresponding standard deviation of 0.874. Although these values are not as good as the original model, their are still acceptable and indicate that model coherence is maintained. The analysis of the plotted results, in figure 4.23a, is in all aspects equal to the one performed for the first model.

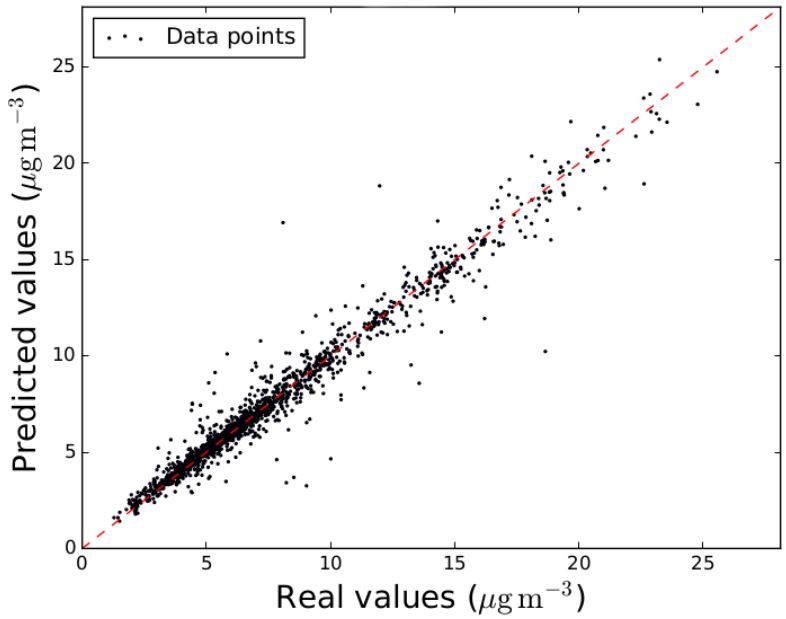
The model using the Savitzky-Golay filter yielded mean absolute percentage errors of 29.650 %, 29.443 % and 29.164 % with the mean value of 29.413 %. and the standard deviation of 0.244 indicating and accurate and coherent model. An analy-

Table 4.11: Dickey-Fuller test results for PM2.5 signals.

	Observations	Test value	P-value	Stationary
Original	8774	-10.69194	3.71E-19	Yes
7 day filter		-11.35856	9.60E-21	Yes
28 day filter		-10.62376	5.43E-19	Yes
Savitzky-Golay filter		-9.05221	4.81E-15	Yes

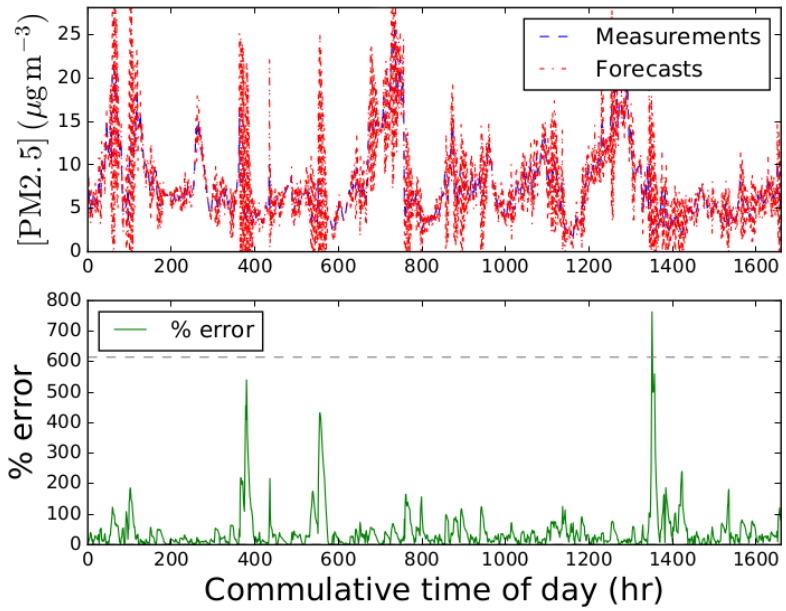


(a)

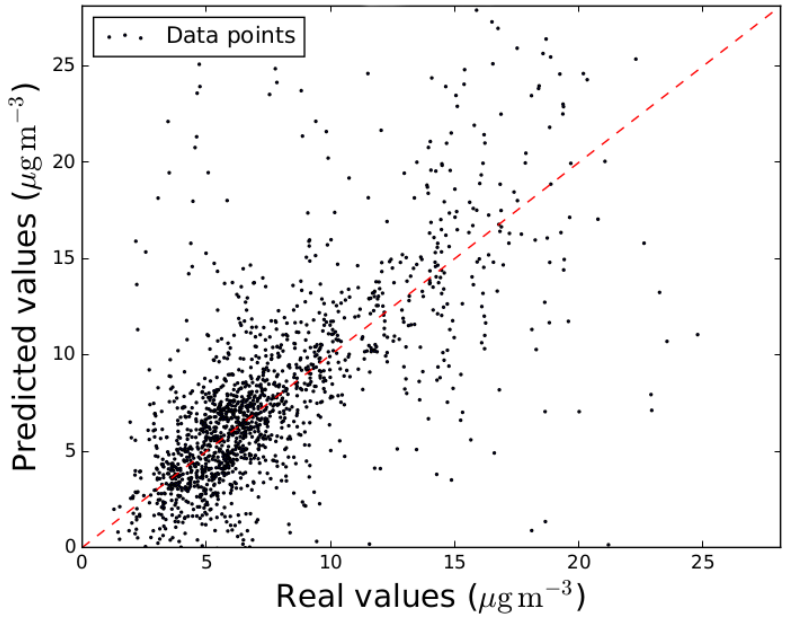


(b)

Figure 4.21: Un-smoothened 24 hour PM2.5 forecasting results for the third iteration. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

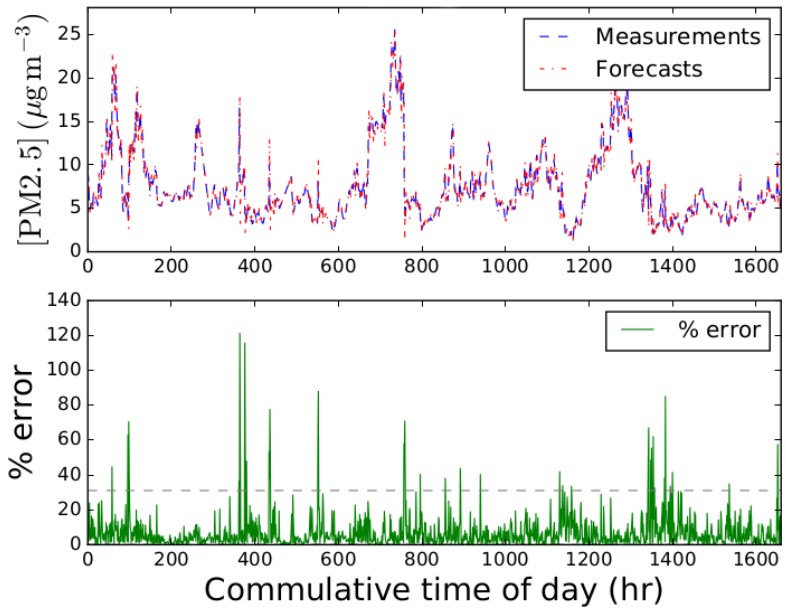


(a)

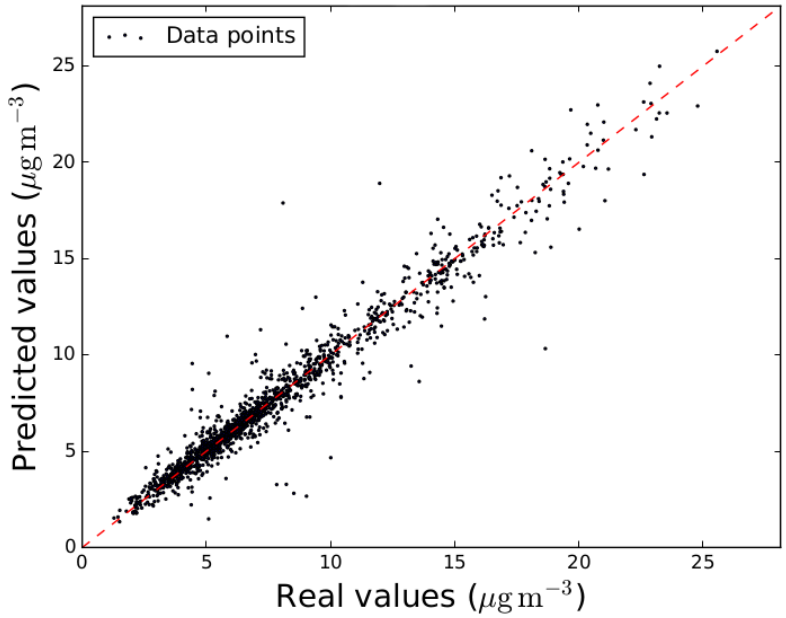


(b)

Figure 4.22: Smoothened 24 hour PM2.5 forecasting results for the third iteration using a moving average with a window of seven days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.



(a)



(b)

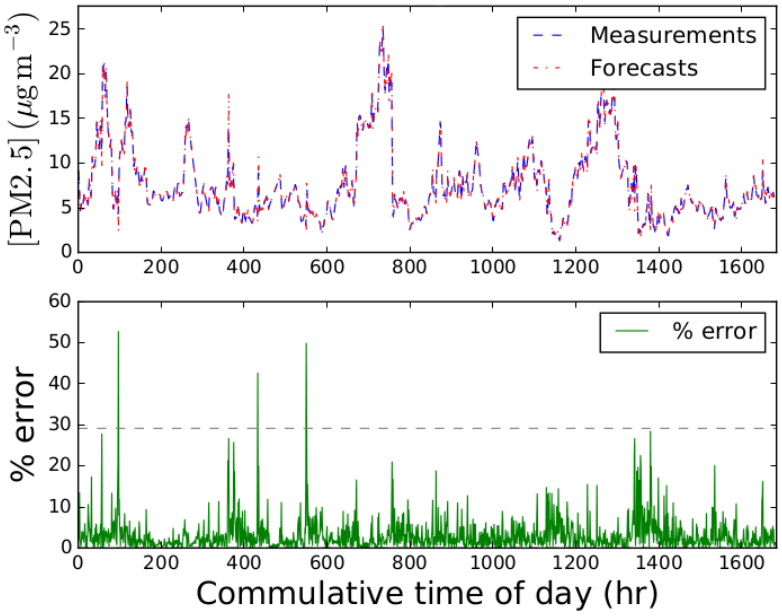
Figure 4.23: Smoothened 24 hour PM2.5 forecasting results for the third iteration using a moving average with a window of 28 *days. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.*

Table 4.12: Machine learning PM2.5 forecasting results. NOTE: all values are in percentages except for the standard deviation (Stdev).

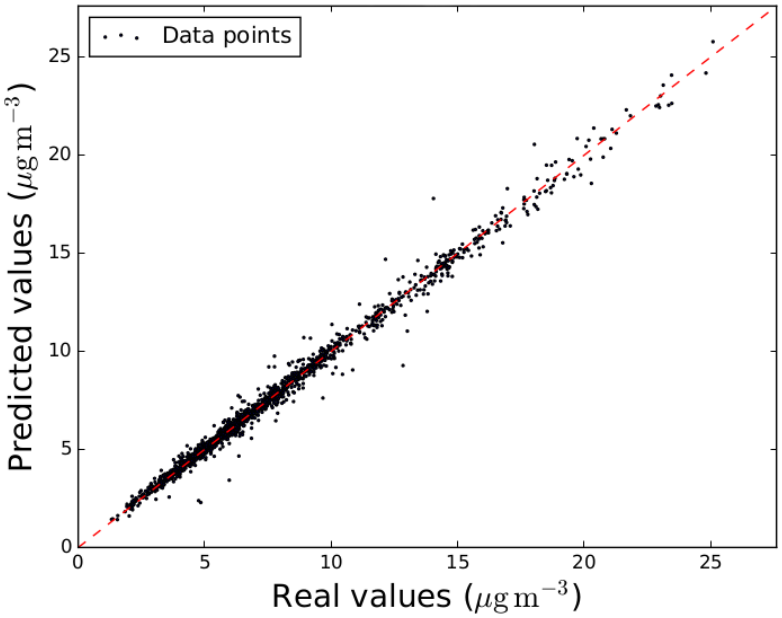
	PM2.5 MAPE				
	Test 1	**Test 2**	**Test 3**	**Mean**	**Stdev**
Original	30.880	30.649	30.369	30.632	0.256
7 day filter	625.096	612.060	614.568	617.241	6.9168
28 day filter	32.764	31.552	31.066	31.794	0.874
Savitzky-Golay	29.650	29.443	29.164	29.420	0.244

sis of the plots, presented in figure 4.24, concludes that the highest individual error barely surpassed the 50 % mark and that there is a very good overlapping of the data points with the quadrant bisection indicating an accurate and coherent model.

A comparative analysis, presented in table 4.12, indicates readily that the model based on the seven day moving average filter lacks accuracy and is not viable in any way. Between the other three models, they produced very similar results with the 28 day moving average filter being the worst. The model based on the Savitzky-Golay filter managed to improve the accuracy of the original model. In addition, this model was able to reduce the highest individual error from both the original and the 28 day moving average models by half.



(a)



(b)

Figure 4.24: Smoothened 24 hour PM2.5 forecasting results for the third iteration using 3^{rd} degree polynomial Savitzky-Golay filter. a) Individual error plot alongside with the MAPE (gray dashed line). b) Correlation dispersion plot.

Conclusions

One of the greatest problems the human population all around the world faces today is the increasing danger posed by air pollution. Compounds such as carbon monoxide, nitrogen dioxide, tropospheric ozone, sulfur dioxide and particulate matter can cause severe health effects and have always been present in our planet's atmosphere, although in low concentrations. However, starting with the dawn of the industrial age, their airborne concentrations have drastically increased creating an even greater threat to our health.

Airborne pollution levels cannot be easily decreased and require a long time to do so and the collective cooperation of the people and nations alike. That said, one possible way to mitigate the problem and reduce the negative impact caused by atmospheric pollution in our societies lies in trying to find ways to accurately and consistently forecast the airborne concentrations of hazardous compounds and particles to pro-actively enable people to defend themselves from the dangers they pose.

This is no easy task given the chaotic nature of the Earth's atmosphere. The complexity of the equations necessary to model only one of the interest compounds stated before is immense and requires extremely powerful computers not only to perform the calculations bu also to derive the equations themselves. And because they are very sensitive to the starting conditions the final models are usually very limited. Because of this difficulty, data driven approaches have been suggested as a viable mean to accurately forecast these concentrations within a fraction of the time and using much less processing power.

In this project a machine learning algorithm based on multiple linear regression models was develop with the aim at forecasting individually each of the before

mentioned pollutants. Also, three different filter types, two based on a moving average and the other being the Savitzky-Golay filter, were applied to each data set and each to account for the effect of signal smoothing on the forecasts. The resulting models were evaluated using the mean absolute relative average as a measure of performance. Table 5.1 depicts MAPE values obtained by the trained models using differently filtered signals.

The first conclusion to be drawn is relative to the improvement, or lack there of, that signal filtering created upon the original model. Overall, all models gained a slight improve in accuracy when the Savitzky-Golay filter was applied prior to training and testing the algorithm. And since the filter itself is not very complex, its application on a dynamic-real-time forecasting system seems viable and necessary since it also managed to severely decrease the size of individual errors, whose measure is uncounted by the MAPE value.

Regarding the moving average based models, they yielded very different results for each pollutant which indicates that the filter is highly dependent on the original signal shape and requires a greater degree of fineness to implement and are less forgiving. For carbon monoxide, tropospheric ozone, and sulfur dioxide both moving average filters yielded worse results than the application of the original signal, meaning they are not necessary to implement in the final algorithm. For particulate matter, the seven day filter severely decreased performance while the 28 day window slightly increased it for PM10 and equaled it for PM2.5 forecasting. Finally, for nitrogen dioxide, the 28 day filter managed to produce a better model being surpassed only by the seven day filter, however since MAPE values obtained while forecasting this pollutant still remain very high all models can be discarded for using in a real-scenario.

One important fact to take notice is that results for CO and NO_2 forecasting are equal up to the third decimal figure which, since the data sets used for training the model were obviously different, means that the lower error limit imposed by the bias-variance dilemma of the model while forecasting both these concentrations was achieved. This implies that these pollutants can now be forecasted real-time without the need for further model improvement.

With MAPE values under 30 %, the models built for forecasting both types of particles are already very useful and capable of being placed in action while still leaving room for further improvement. And although the same might be said for

Table 5.1: Performance metrics summary.

	MAPE					
	CO	**NO2**	**O3**	**SO2**	**PM10**	**PM2.5**
Original	7.719	175.274	7.719	43.099	27.539	30.632
7 day filter	10.302	160.195	10.302	160.195	497.121	617.241
28 day filter	13.227	165.809	13.227	884.629	27.090	31.793
Savitzky-Golay	7.446	170.742	7.446	41.471	26.451	29.419

the sulfur dioxide model, its MAPE value of over 40 % still has a high enough margin of error for the model to cause doubts if put into practice on real-time data.

To summarize, the application of Savitzky-Golay filter managed to consistently improve forecasts for all compounds while the moving average filters proved to be very erratic in doing so. Also, good models (MAPE $<\approx 10\,\%$) were developed for carbon monoxide and tropospheric ozone forecasting while adequate models ($10\,\% <\approx$ MAPE $<\approx 45\,\%$) were produced for particulate matter and sulfur dioxide forecasting and no model model produced was able to accurately forecast nitrogen dioxide concentrations.

Future remarks

This project focused only on numerical procedures and did not take into account correlation between the several case variables and yet it was able to produce models to accurately forecast carbon monoxide and tropospheric ozone concentrations without the need for much improvement.

The next step lies in trying to improve the other developed models by adding a degree of correlation between other known variables. As such blind source separation techniques such as principal and independent component analysis seem the logical next step to try and find the existing correlations that the six studied pollutants have with each other in order to manage to create indirect forecasting models based on carbon monoxide and ozone direct predictions. Not only that, but also the inclusion of variables such as temperature, humidity, wind direction and speed that are easier to forecast might also facilitate the creation of indirect forecasting models.

Other than that, existent models (with the exception of the nitrogen dioxide model) can already be placed in use in real scenarios to verify if they are able to maintain the same level of performance that they had during tests in a real situation. Also, withing their error margins, alerts could be issued to the general population in case of a suspected rise in any of the pollutants airborne concentrations.

Bibliography

[1] World Health Organization. Ambient (outdoor) air quality and health, 2016.

[2] L. Wang, B. Zhong, S. Vardoulakis, F. Zhang, E. Pilot, and Y. Li. Air Quality Strategies on Public Health and Health Equity in Europe — A Systematic Review. *Int. J. Environ. Res. Public. Health.*, 13(12):1196–1220, 2016.

[3] Y. Chu, Y. Liu, X. Li, Z. Liu, H. Lu, Y. Lu, Z. Mao, X. Chen, N. Li, M. Ren, F. Liu, L. Tian, Z. Zhu, and H. Xiang. A review on predicting ground PM2.5 concentration using satellite aerosol optical depth. *Atmosphere*, 7(10):129–154, 2016.

[4] K. Katsouyanni. Ambient air pollution and health. *Brit. Med. Bul.*, 68(1): 143–156, 2003.

[5] T. Boden, G. Marland, and R. Andres. Global, Regional, and National Fossil-Fuel CO2 Emissions. Technical report, Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, U.S. Department of Energy, Tennessee , U.S.A., 2015.

[6] S. Gautam, A. Yadav, C. Tsai, and P. Kumar. A review on recent progress in observations, sources, classification and regulations of PM2.5 in Asian environments. *Environ. Sci. Pollut. R.*, 23(21):21165–21175, 2016.

[7] S. Breitner, A. Schneider, R. Devlin, C. Ward-caviness, D. Diaz-sanchez, L. Neas, W. Cascio, A. Peters, E. Hauser, S. Shah, and W. Kraus. Associations among plasma metabolite levels and short-term exposure to PM 2.5 and ozone in a cardiac catheterization cohort. *Environ. Int.*, 97:76–84, 2016.

[8] European Environment Agency. State and Outlook 2015 the European Environment. Technical report, EEA2015:212, 2015.

[9] 1979 Convention on long-range transboundary air pollution. Technical report, United Nations Economic Commission for Europe, Geneva, 1979.

[10] United Nations Economic Commission for Europe. Decision 2010/18 - Long-term strategy for the convention on long-range transboundary air pollution and action plan for its implementation. Technical report, United Nations Economic Commission for Europe, Geneva, 2010.

[11] European Environment Agency. Atmosphere Monitoring — Copernicus, 2014. URL `http://copernicus.eu/main/atmosphere-monitoring; Accessed:2017-01-30` .

[12] L. Prockop and R. Chichkova. Carbon monoxide intoxication: an updated review. *J. Neurol. Sci.*, 262(1-2):122–30, 2007.

[13] S. Omaye. Metabolic modulation of carbon monoxide toxicity. *Toxicology*, 180(2):139–150, 2002.

[14] G. Kikuchi, T. Yoshida, and M. Noguchi. Heme oxygenase and heme degradation. *Biochem. Bioph. Res. Co.*, 338(1):558–567, 2005.

[15] J. Evans, F. Niemevz, G. Buldain, and P. de Montellano. Isoporphyrin intermediate in heme oxygenase catalysis. Oxidation of alpha-meso-phenylheme. *J. Biol. Chem.*, 283(28):19530–9, 2008.

[16] J. Berg, J. Tymoczko, and L. Stryer. *Biochemistry*. W.H. Freeman, 2002.

[17] L. Kao and K. Nañagas. Toxicity Associated with Carbon Monoxide. *Clin. Lab. Med.*, 26(1):99–125, mar 2006.

[18] M. Goldstein, J. Raphael, J. Korach, M. Jars-Guincestre, C. Chastang, and C. Harboun. Carbon monoxide poisoning. *J. Emerg. Nurs.*, 34(6):538–42, dec 2008.

[19] G. Vásquez, X. Ji, C. Fronticelli, and G. Gilliland. Human Carboxyhemoglobin at 2.2 A Resolution: Structure and Solvent Comparisons of R-State, R2-State and T-State Hemoglobins. *Acta Crystallogr. D.*, 54(3):355–366, 1998.

[20] N. Buckley, G. Isbister, B. Stokes, and D. Juurlink. Hyperbaric Oxygen for Carbon Monoxide Poisoning. *Toxicol. Rev.*, 24(2):75–92, 2005.

[21] World Health Organization. Chapter 7.1 Nitrogen dioxide General description. In *Air quality guidelines*, chapter 7.1. Copenhagen, 2nd edition, 2000.

[22] P. Crutzen, M. Molina, and F. Rowland. The Nobel Prize in Chemistry 1995. URL `http://www.nobelprize.org/nobel_prizes/chemistry/laureates/1995/;Accessed:2017-03-20` .

[23] US EPA. Integrated ccience assessment for oxides of nitrogen – health criteria (2016 final report). Technical report, U. S. EPA, Washington, DC, 2016.

[24] C. Weschler, J. Wells, D. Poppendieck, H. Hubbard, and T. Pearce. Workgroup report: Indoor chemistry and health. *Environ. Health Persp.*, 114(3):442–6, 2006.

[25] T. Devasagayam, J. Tilak, K. Boloor, Ke. Sane, S. Ghaskadbi, and R. Lele. Free Radicals and Antioxidants in Human Health : Current Status and Future Prospects. *J. Assoc. Physicians India*, 52(4):794–804, 2004.

[26] B. Halliwell and S. Chirico. Lipid peroxidation: its mechanism, measurement, and significance. *Am. J. Clin. Nutr.*, 57(5):715S–724S; discussion 724S–725S, 1993.

[27] H. Cai and D. Harrison. Endothelial Dysfunction in Cardiovascular Diseases: The Role of Oxidant Stress. *Circ. Res.*, 87(10):840–844, 2000.

[28] T. Berhanu, J. Savarino, S. Bhattacharya, and W. Vicars. 17O excess transfer during the NO2 + O3 → NO3 + O2 reaction. *J. Chem. Phys.*, 136(4):044311, jan 2012.

[29] B. Ayres, H. Allen, D. Draper, S. Brown, R. Wild, J. Jimenez, D. Day, P. Campuzano-Jost, W. Hu, J. de Gouw, A. Koss, R. Cohen, K. Duffey, P. Romer, K. Baumann, E. Edgerton, S. Takahama, J. Thornton, B. Lee, F. Lopez-Hilfiker, C. Mohr, P Wennberg, T. Nguyen, A. Teng, A. Goldstein, K. Olson, and J. Fry. Organic nitrate aerosol formation via NO3 + biogenic

volatile organic compounds in the southeastern United States. *Atmos. Chem. Phys.*, 15:13377–13392, 2015.

[30] World Health Organization. Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide. Technical report, World Health Organization, 2005.

[31] M. Kirschner. Ozone. In *Ullmann's Encyclopedia of Industrial Chemistry*. Wiley-VCH Verlag GmbH and Co. KGaA, Weinheim, Germany, 2000.

[32] R. Ackermann, S. Aggarwal, J. Dixon, A. Fitzgerald, D. Hanrahan, Gordon A. Hughes, Arundhati Kunte, Magda Lovei, Kseniya Lvovsky, and Anil H. Somani. Ground-Level Ozone. In *Pollution prevention and abatement handbook*, chapter 10. World Bank Group, Washington, D.C., 1st edition, 1998.

[33] C. Reeves, S. Penkett, S. Bauguitte, K. Law, M. Evans, B. Bandy, P. Monks, G. Edwards, G. Phillips, H. Barjat, J. Kent, K. Dewey, S. Schmitgen, and D. Kley. Potential for photochemical ozone formation in the troposphere over the North Atlantic as derived from aircraft observations during ACSOE. *J. Geophys. Res.-Atmos.*, 107(D23):ACH 14–1–ACH 14–14, 2002.

[34] B. Weinhold. Ozone nation: EPA Standard Panned by the People. *Environ. Health Persp.*, 116(7):302–305, 2008.

[35] M. Jenkin and K. Clemitshaw. Ozone and other secondary photochemical pollutants: chemical processes governing their formation in the planetary boundary layer. *Atmos. Environ.*, 34(16):2499–2527, 2000.

[36] S. Chapman. *A Theory of Upper-Atmospheric Ozone*. Edward Stanford, 1930.

[37] B. Finlayson-Pitts and J. Pitts Jr. *Chemistry of the upper and lower atmosphere: theory, experiments, and applications*. Academic press, 1999.

[38] L. Smith. Oxygen, oxysterols, ouabain, and ozone: a cautionary tale. *Free radical Bio. Med.*, 37(3):318–324, 2004.

[39] N. Schlager, D. Newton, and Jayne. Weisblatt. *Encyclopedia of Chemical Compounds*. Thomson Gale, Farmington Hills, 1st edition, 2006.

[40] Z. Meng, G. Qin, and B. Zhang. DNA damage in mice treated with sulfur dioxide by inhalation. *Environ. Mol. Mutagen.*, 46(3):150–155, 2005.

[41] Z. Meng, G. Qin, B. Zhang, H. Geng, Q. Bai, W. Bai, and C. Liu. Oxidative damage of sulfur dioxide inhalation on lungs and hearts of mice. *Environ. Res.*, 93(3):285–292, 2003.

[42] Z. Meng and L. Zhang. Chromosomal aberrations and sister-chromatid exchanges in lymphocytes of workers exposed to sulphur dioxide. *Mutat. Res.-Genet. Tox.*, 241(1):15–20, 1990.

[43] J. Yadav and V. Kaushik. Effect of sulphur dioxide exposure on human chromosomes. *Mutat. Res.-Envir. Tox.*, 359(1):25–29, 1996.

[44] R. Li, Z. Meng, and J. Xie. Effects of sulfur dioxide derivatives on four asthma-related gene expressions in human bronchial epithelial cells. *Tox. Lett.*, 175 (1-3):71–81, 2007.

[45] US EPA. Particulate Matter (PM) Basics. URL `https://www.epa.gov/pm-pollution/particulate-matter-pm-basics;Accessed:2017-06-05` .

[46] H. Macintyre, C. Heaviside, L. Neal, P. Agnew, J. Thornes, and S. Vardoulakis. Mortality and emergency hospitalizations associated with atmospheric particulate matter episodes across the UK in spring 2014. *Environ. Int.*, 97:108–116, 2016.

[47] R. Xie, C. Sabel, X. Lu, W. Zhu, H. Kan, C. Nielsen, and H. Wang. Long-term trend and spatial pattern of PM2.5 induced premature mortality in China. *Environ. Int.*, 97:180–186, 2016.

[48] H. Taheri Shahraiyni and S. Sodoudi. Statistical Modeling Approaches for PM10 Prediction in Urban Areas; A Review of 21st-Century Studies. *Atmosphere*, 7(2):15, 2016.

[49] M. Matti Maricq. Chemical characterization of particulate emissions from diesel engines: A review. *J. Aerosol Sci.*, 38(11):1079–1118, 2007.

[50] F. Amato, M. Pandolfi, T. Moreno, M. Furger, J. Pey, A. Alastuey, N. Bukowiecki, A. Prevot, U. Baltensperger, and X. Querol. Sources and variability of inhalable road dust particles in three European cities. *Atmos. Environ.*, 45(37):6777–6787, 2011.

[51] P. Lenschow. Some ideas about the sources of PM10. *Atmos. Environ.*, 35: 23–33, 2001.

[52] J. Keary, S. Jennings, T. O'Connor, B. McManus, and M. Lee. PM10 Concentration Measurements in Dublin City. In *Urban Air Quality: Monitoring and Modelling*, pages 3–18. Springer Netherlands, Dordrecht, 1998.

[53] P. de Mattos Neto, G. Cavalcanti, F. Madeiro, and T. Ferreira. An Approach to Improve the Performance of PM Forecasters. *PLoS ONE*, 10(9):138–507, 2015.

[54] L. Fajersztajn, M. Veras, L. Barrozo, and P. Saldiva. Air pollution: a potentially modifiable risk factor for lung cancer. *Nat. Rev. Cancer*, 13(9):674–678, 2013.

[55] J. Feng and W. Yang. Effects of Particulate Air Pollution on Cardiovascular Health: A Population Health Risk Assessment. *PLoS ONE*, 7(3):e33385, 2012.

[56] M. Kleinman. Central nervous system effects of ambient particulate matter: the role of oxidative stress and inflammation final report. Technical report, 2014.

[57] M. Hilbert and P. Lopez. The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025):60–65, 2011.

[58] D. Boyd and K. Crawford. Six Provocations for Big Data. *A decade in internet time: Symposium on the dynamics of the internet and society*, 21, 2011.

[59] M. Hilbert. Big Data for Development: A Review of Promises and Challenges. *Dev. Policy Rev.*, 34(1):135–174, 2016.

[60] J. Manyika, B. Brown, J. Bughin, A. Byers, M. Chui, R. Dobbs, and C. Roxburgh. Big Data: The next frontier for innovation, competition, and productivity. Technical report, 2011.

[61] M. Spiegel and L. Stephens. Analysis of Time Series. In *Theory and Problems of Statistics*, chapter 18. McGraw-Hill, New York, 3rd edition, 1998.

[62] D. Dickey and W. Fuller. Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *J. Am. Stat. Assoc.*, 74(366):427, 1979.

[63] W. Enders. *Applied Econometric Time Series*. Wiley, New York, third edition, 2010.

[64] M. Spiegel and L. Stephens. Curve Fitting and the Method of least Squares. In *Theory and Problems of Statistics*, chapter 13. McGaw-Hill, New York, 3rd edition, 1998.

[65] J. Hamilton. *Time series analysis*. Princeton University Press, Princeton, New Jersey, 1st edition, 1994.

[66] M. Hatanaka. *Time-series-based econometrics : unit roots and co-integrations*. Oxford University Press, Oxford, 1st edition, 1996.

[67] Ph. Guest. *Numerical methods of curve fitting*. Cambridge University Press, Cambridge, 1st paperb edition, 2013.

[68] A. Oppenheim and R. Schafer. *Digital signal processing*. Prentice-Hall, 1st edition, 1975.

[69] A. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

[70] P. Harrington. *Machine Learning in Action*. Manning Publications, New York, 1st edition, 2010.

[71] R. Kohavi and F. Provost. Glossary of Terms Journal of Machine Learning. URL `http://robotics.stanford.edu/$\sim$ronnyk/glossary.html; Accessed:2017-04-12` .

[72] M. Mohri, Af. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT Press, Cambridge, 1st edition, 2012.

[73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, Ro. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2012.

[74] J. Brownlee. Supervised and Unsupervised Machine Learning Algorithms - Machine Learning Mastery. URL `http://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/;Accessed:2017-05-15` .

[75] P. Stalph. Introduction to Function Approximation and Regression. In *Analysis and Design of Machine Learning Techniques*, chapter 2, pages 11–28. Springer Fachmedien Wiesbaden, Wiesbaden, 1st edition, 2014.

[76] E. Alpaydin. *Introduction to machine learning*. MIT Press, Cambridge, 1st edition, 2010.

[77] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, Stanford, 2nd edition, 2013.

[78] C. Bishop. *Pattern Recognition and Machine Learning*. Singapure, 1st edition, 2006.

[79] D. Lindley. Regression and correlation analysis, 1987.

[80] H. Seltman. Experimental Design and Analysis. Technical report, Carnegie Mellon University, Pittsburgh, 2015.

[81] D. Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, New York, 1st edition, 2009.

[82] N. Bourbaki. *Topological vector spaces*. Springer, New York, 2nd edition, 1987.

[83] A. Stuart and K. Ord. *Kendall's advanced theory of statistics*. John Wiley and Sons, London, 6th edition, 1994.

[84] M. Murty and V. Devi. Bayes Classifier. In *Pattern Recognition: An Algorithmic Approach*, chapter 4, pages 86–102. Springer and Universities Press (India), London, 1st edition, 2011.

[85] A. Thakkar. Molecular symmetry. In *Quantum Chemistry*, chapter 1, pages 1–10. Morgan and Claypool Publishers, San Rafael, 1st edition, 2014.

[86] Z. Pozun, K. Hansen, D. Sheppard, M. Rupp, K. Müller, and G. Henkelman. Optimizing transition states via kernel-based machine learning. *J. Chem. Phys.*, 136(17):174101, 2012.

[87] C. Cortes and V. Vapnik. Support-vector networks. *Mach. learn.*, 20(3):273–297, 1995.

[88] W. Press. *Numerical recipes: the art of scientific computing.* Cambridge University Press, 2007.

[89] N. Altman. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.*, 46(3):175–185, 1992.

[90] P. Jaskowiak and R. Campello. Comparing Correlation Coefficients as Dissimilarity Measures for Cancer Classification in Gene Expression Data. *BSB'2011*, pages 1–8, 2011.

[91] D. Coomans and D. Massart. Alternative k-nearest neighbour rules in supervised pattern recognition. *Anal. Chem. Acta*, 136:15–27, 1982.

[92] B. Everitt, S. Landau, M. Leese, and D. Stahl. Miscellaneous Clustering Methods. In *Cluster Analysis*, chapter 8, pages 215–255. John Wiley and Sons, London, 5th edition, 2011.

[93] K. Lipkowitz and D. Boyd. *Reviews in computational chemistry, volume 18.* John Wiley and Sons, New Jersey, 1st edition, 2002.

[94] A. Rencher and W. Christensen. Multivariate Regression. In *Methods of Multivariate Analysis*, chapter 10, pages 339–383. John Wiley and Sons, New Jersey, 3rd edition, 2012.

[95] J. Armstrong. Illusions in regression analysis. *int. J. Forecasting*, 28(3):689–694, 2012.

[96] J. Kenney and E. Keeping. *Mathematics of statistics, part 1.* D. Van Nostrand Company, Princeton, 3rd edition, 1954.

[97] S. Stigler. Gauss and the Invention of Least Squares. *Ann. Stat.*, 9(3):465–474, 1981.

[98] A. Charnes, E. Frome, and P. Yu. The Equivalence of Generalized Least Squares and Maximum Likelihood Estimates in the Exponential Family. *J. Am. Stat. Assoc.*, 71(353):169–171, 1976.

[99] A. Goldberger. *Econometric theory*. John Wiley and Sons, New Jersey, 1st edition, 1964.

[100] C. Lawson and R. Hanson. *Solving least squares problems*. Prentice-Hall, New Jersey, 1st edition, 1974.

[101] N. Draper and H. Smith. *Applied regression analysis*. John Wiley and Sons, New Jersey, 1st edition, 1998.

[102] K. Mardia, J. Kent, and J. Bibby. *Multivariate analysis*. Academic Press, 1979.

[103] D. Cox. The regression analysis of binary sequences (with discussion). *J. Roy. Stat. Soc. B.*, 20:215–242, 1958.

[104] S. Walker and D. Duncan. Estimation of the Probability of an Event as a Function of Several Independent Variables. *Biometrika*, 54(1/2):167, jun 1967.

[105] E. Weisstein. Sigmoid Function, . URL `http://mathworld.wolfram.com/SigmoidFunction.html;Accessed:2017-05-16` .

[106] W. Cleveland. Robust Locally Weighted Regression and Smoothing Scatterplots. *J. Amer. Statist. Assoc.*, 74(368):829, 1979.

[107] W. Cleveland and S. Devlin. Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. *J. Amer. Statist. Assoc.*, 83(403):596, 1988.

[108] B. Schoolkopf, K. Tsuda, and J. Vert. *Kernel methods in computational biology*. MIT Press, Cambridge, 1st edition, 2004.

[109] S. Geman, E. Bienenstock, and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Comput.*, 4(1):1–58, 1992.

[110] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*, volume 103 of *Springer Texts in Statistics*. Springer, New York, 1st edition, 2013.

[111] Y. Wang, D. Miller, and R. Clarke. Approaches to working in high-dimensional data spaces: gene expression microarrays. *Brit. J. Cancer.*, 98(6):1023–8, 2008.

[112] A. Tikhonov and V. Arsenin. *Solutions of ill-posed problems*. Winston, New York, 1st edition, 1977.

[113] A. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[114] R. Tibshiranit. Regression Shrinkage and Selection via the Lasso. *J. Roy. Stat. Soc. B Met.*, 58(1):267–288, 1996.

[115] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 2(12):1137–1143, 1995.

[116] P. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, London, 1st edition, 1982.

[117] G. Seni, J. Elder, and H. Liu. *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan and Claypool Publishers, 1st edition, 2010.

[118] G. McLachlan, C. Ambroise, and K. Do. *Analyzing microarray gene expression data*. John Wiley and Sons, New Jersey, 1st edition, 2004.

[119] S. Ben Taieb, G. Bontempi, Amir F. Atiya, and A. Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Syst. Appl.*, 39(8):7067–7083, 2012.

[120] A. Sfetsos. Time Series Forecasting of Hourly PM10 Using Localized Linear Models. *J. Software Eng. Appl.*, 03(4):374–383, 2010.

[121] C. Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *J. Oper. Res. Soc.*, 66:1352–1362, 2015.

[122] V. Estivill-Castro. Why so many clustering algorithms. *SIGKDD Explor.*, 4 (1):65–75, 2002.

[123] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6, 2000.

[124] P. Pudil and J. Novovičová. Novel Methods for Feature Subset Selection with Respect to Problem Knowledge. In *Feature Extraction, Construction and Selection*, chapter 7, pages 101–116. Springer, Boston, 1st edition, 1998.

[125] H. Zare, P. Shooshtari, A. Gupta, and R. Brinkman. Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC Bioinformatics*, 11(1):403, 2010.

[126] H. Kriegel, E. Schubert, and A. Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowl. Inf. Syst.*, pages 1–31, 2016.

[127] D. Ketchen Jr. and C. Shook. The application of cluster analysis in Strategic Management Research: An analysis and critique. *Strateg. Manage J.*, 17(6): 441–458, 1996.

[128] L. Rokach and O. Maimon. Clustering Methods. In *Data mining and knowledge discovery handbook*, chapter 15, pages 321—-352. Springer, Tel-Aviv, 2005.

[129] R. Brown and Y. Martin. Use of Structure-Activity Data To Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection. *J. Chem. Inf. Comp. Sci.*, 36(3):572–584, 1996.

[130] L. Tao, F. Zhu, C. Qin, C. Zhang, S. Chen, P. Zhang, C. Zhang, C. Tan, C. Gao, Z. Chen, Y. Jiang, and Y. Chen. Clustered Distribution of Natural Product Leads of Drugs in the Chemical Space as Influenced by the Privileged Target-Sites. *Sci. Rep.*, 5(1):9325, 2015.

[131] H. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *WIREs Data. Mining. Knowl. Discov.*, 1(3):231–240, 2011.

[132] X. Duan, F. Yang, E. Antono, W. Yang, P. Pianetta, S. Ermon, A. Mehta, and Y. Liu. Unsupervised Data Mining in nanoscale X-ray Spectro- Microscopic Study of NdFeB Magnet. *NPG*, 6, 2016.

[133] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD'96*, pages 226–231, 1996.

[134] B. Frey and D. Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315, 2007.

[135] U. Von Luxburg. A Tutorial on Spectral Clustering. Technical report, Max Planck Institute for Biological Cybernetics, Tubingen, 2007.

[136] E. Weisstein. Eigen Decomposition, . URL `http://mathworld.wolfram.com/EigenDecomposition.html;2017-05-28` .

[137] I. Jolliffe. Introduction. In *Principal Component Analysis*, chapter 1, pages 1–9. Springer, New York, 2nd edition, 2002.

[138] I. Jolliffe. Mathematical and Statistical Properties of Sample Principal Components. In *Principal Component Analysis*, chapter 3, pages 29–61. Springer, New York, 2nd edition, 2002.

[139] M. Leznik and C. Tofallis. Estimating Invariant Principal Components Using Diagonal Regression. 2005.

[140] P. Comon. Independent component analysis, A new concept? *Signal Process.*, 36(36):28–314, 1994.

[141] A. Hyvärinen, J. Karhunen, and E. Oja. Independent Component Analysis. In *Independent Component Analysis*, chapter 1. John Wiley and Sons, New York, 1st edition, 2001.

[142] J. Pereira, J. Azevedo, H. Knapik, and H. Burrows. Unsupervised component analysis: PCA, POA and ICA data exploring - connecting the dots. *Spectrochim. Acta A*, 165:69–84, 2016.

[143] J. Demmel and W. Kahan. Accurate Singular Values of Bidiagonal Matrices. *SIAM J. Sci. Comput.*, 11(5):873–912, 1990.

[144] M. Sahidullah and T. Kinnunen. Local spectral variability features for speaker verification. *Digit. Signal Process.*, 50:1–11, 2016.

[145] O. Alter, P. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *P. Natl. Acad. Sci. USA*, 97 (18):10101–6, 2000.

Appendices

PM2.5 and PM10: a case study of Coimbra area in Portugal

João A. Gregório,[*,†,¶] Pedro J. S. B. Caridade,[†] and Carla Gouveia-Caridade[‡]

†*Department of Chemistry, Faculty of Sciences and Technology, University of Coimbra, Coimbra*

‡*SpaceLayer Technologies, Pedro Nunes Institute, Coimbra*

¶*Current address: Coimbra, Portugal*

E-mail: jgregorio@student.uc.pt

Phone: +351 91 237 8769

Abstract

Air pollution is a serious environmental issue impacting negatively human productivity and heath and even causing premature death. "Citizens often breathe air that does not meet standards, with major sequels".[1] Being able to forecast accurately the concentrations of pollutants such as particulate matter is of great importance to enable citizens to act preventive. In this work a machine learning algorithm based on multivariate linear regression is developed and proposed as a method for forecasting PM10 and PM2.5 concentrations. The presented algorithm is shown to able to forecast particulate matter concentrations with a decent degree of accuracy and shows much promise while still leaving room for further improvements.

Keywords: Environmental chemistry, multivariate linear regression, time series forecasting, PM10 forecasting, PM2.5 forecasting, environmental data analysis

Introduction

Air pollution is one of the biggest environmental concerns in present days affecting people all around the world, in developed and developing countries alike. The World Health Organization (WHO) estimates that in 2012 up to 3.7 million premature deaths all around the world could be blamed on air pollution.[2–4] Ambient air pollutants are diverse, but from all the contaminants, a few stand out as the most dangerous to human health such as particulate matter (PM), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. From these, particulate matter, constituted of fine particles from organic and inorganic sources with aerodynamic diameters smaller than $10\,\mu$m (PM10) and than $2.5\,\mu$m (PM2.5), are considered the most aggressive and threatening to human health, affecting more people than any other airborne pollutant.[2,4–7]

The formation of PM can be divided in two main sources[8]: primary sources attributed in urban areas to road traffic, such as, carbonaceous compounds from exhaust emissions[9], re-suspension of road dust[10], and tyre abrasion[11] and combustion processes; and secondary sources ascribed to the condensation of vapours or chemical reactions such as atmospheric oxidation of SO_2 to H_2SO_4, and NO_2 to HNO_3.[12]

Due to their small size, when they are breathed in, fine particles are able to penetrate deep into the lungs and even be absorbed into the blood stream causing damage to the organism. The level of injury they can cause varies widely depending on their concentration and type.[2,13,14] Given the nature of their absorption, the damage they cause is mostly focused on the respiratory system, although the cardiovascular and neurological systems can also be affected by proxy if the particles are extremely small and hazardous.[15–17] Some of the less severe short-term effects include irritation of the mucous areas like eyes, nose and throat, headaches and nausea which disappear with time. However chronic exposure to high levels is also linked to more serious conditions and can cause upper respiratory infections like bronchitis and emphysema.[2,13] Regarding long-term effects PM exposure is also linked to chronic respiratory diseases such as asthma and lung cancer[15], cardiovascular ailments[16,18]

and brain damage.[17]

Considering the danger posed by particulate matter, it is necessary to survey and monitor its ambient levels in the atmosphere.[4,19] However, since particulate matter is an extremely heterogeneous mixture it becomes very difficult to measure and quantify. This task was made easier by standardizing fine particles with diameters equal or smaller to $10\,\mu$m as the official measure of ambient particle pollution.[20] Also, the increase in emissions and recognition of the heightened dangers of fine particles with an average diameter equal or smaller than $2.5\,\mu$m made it necessary to also pay attention to these particles.[4,21,22] Having these facts into consideration, the ability to accurately predict and forecast PM10 and PM2.5 values is of great interest for public health because it would give citizens a good measure of ambient levels of pollutants and would enable the general population to take preventive actions.

The traditional approach to forecasting environmental variables implies the construction of deterministic models which require extensive knowledge of parameters such as air current flows, particle diffusion and chemical reactions.[23] The drawbacks of these approaches lie in the data acquisition and model construction processes. Moreover, usable data for all necessary parameters is hard to collect and even if the necessary data is available the algorithm construction and refinement process is lengthy, very demanding and often produces inaccurate models given to the chaotic nature of the atmosphere.[24] In recent years, the application of Machine Learning Algorithms (MLA) has surfaced and gained reputation as a viable alternative for modeling and forecasting time series (TS) data such as PM10 and PM2.5 concentrations.[13,25–28] The advantage of these type of algorithms lies in the fact that they are capable of capturing and finding the underlying patterns hidden in data and use them for forecasting without the need to make any prior assumptions. This means that not only is the model building process less demanding than that of the traditional approach but also that the model application to new data is very straightforward and extremely fast by comparison.[27,28]

This work aims at finding out if a simple forecasting methodology, like one based of

linear regression, has enough viability to accurately predict near-future PM10 and PM2.5 concentrations in the air. The need for a simple method derives from the fact that it is intended for incorporation in a larger system which includes information dissemination.[29]

Experimental section

This study relies on official satellite data, taken over the city of Coimbra, Portugal. Data was provided by the European Space Agency (ESA) under the Copernicus - The European Earth Observation Programme[30], hence it is considered as empirically correct. The central point of all the measurements is "Instituto Pedro Nunes" (IPN) located at 40.192169N -8.414162W. Since the data was taken via satellite no consideration is given to the geography and climate of the city itself. Also, measurements were done on an hourly basis from 01-10-2016 up until 30-09-2017.

Before training the MLA on the original data set, a few pretreatment steps were required as seen in figure 1. Firstly, the data set was incomplete and had a few data gaps that made it impossible to properly use any time-series forecasting model based on machine learning principles. However, since the gaps were few and short in size a placewise linear interpolation model was hypothesised to be enough to fill them. This simple model connects the two known adjacent points, (x_1, y_1) and (x_2, y_2), with a straight line and allows to determine a set of i values in-between them:

$$y_i = y_1 + (x_i - x_1)\frac{y_2 - y_1}{x_2 - x_1} \tag{1}$$

Despite the gaps being few and small, time series properties cannot be changed drastically by data inference without it compromising future results. To confirm that the the data set properties were maintained after the interpolation, the Dickey-Fuller test[31] for stationarity was applied on the original data set and on the continuous data set:

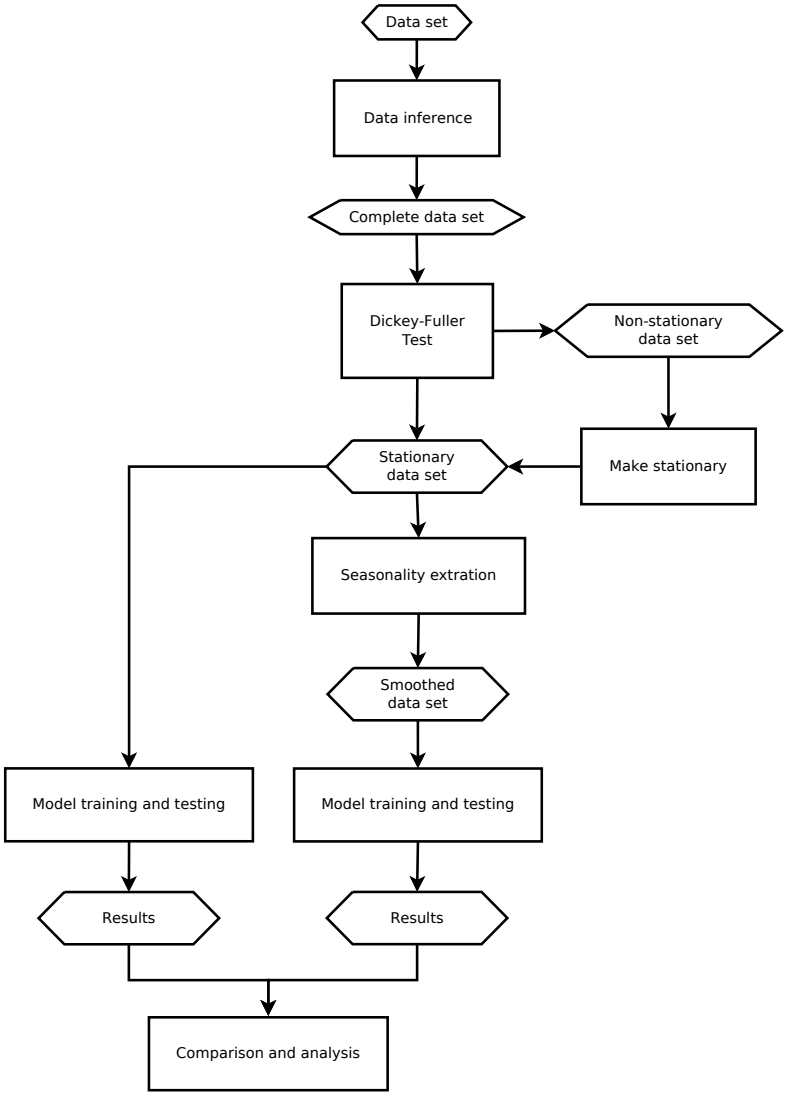


Figure 1: Data processing flowchart and algorithm application.

$$y_t - y_{t-1} = (\rho - 1)y_{t-1} + u_t \tag{2}$$

where y_t is the variable of interest, t is the time index, u_t is the error term and ρ is the test coefficient. If $\rho = 1$ the series would be completely non-stationary. The greater the difference between zero and $\rho - 1$ is, the greater the certainty with which the series can be called stationary. This also allowed to check for the TS stationarity since any regression MLA can only be applied on stationary data.[32] If the time-series is not stationarity, a detrending process has to be done before being able to train the MLA. Only after these steps can the training process of the MLA take place.

The choice of a multiple-input multiple-output linear regression model is mostly based on the fact that it does not have the drawbacks of multiple linear regression models such as the accumulation of errors along the forecasting horizon in recursive models and the conditional independence assumption in direct models[26]. These models are based of vector-valued functions such as the one shown in equation 3.

$$\mathbf{Y}^{(i)}[y_{t+1}, ..., y_{t+H}] = \mathbf{X}^{(i)}(y_t, ..., y_{t-d+1}) \cdot \mathbf{C}_{(d+1)\times i} + \mathbf{w}_{H\times i} \tag{3}$$

In equation 3, $\mathbf{Y}$ and $\mathbf{X}$ are the output and input vectors, respectively, for a forecast of H steps ahead using d input values where i represents the size of the training set, or in other words the total number of equations for which the coefficients matrix, $\mathbf{C}$ is going to be minimized via a least squares approach. Finally, $\mathbf{w}$ is the noise matrix.[26,33,34]

The least square minimization process is done by minimizing the sum of the squares elements on the diagonal of the residual sum of squares and cross products matrices[35]:

$$C = (X'X)^{-1}(X'Y) \tag{4}$$

which gives the least possible trace for the coefficients matrix as well as minimizes the generalized variance of the system. The final coefficients can be inputed in a simplified

vector equation similar to equation 3 alongside the input vector in order to produce the forecasts:

$$\mathbf{Y}[y_{t+1}, ..., y_{t+H}] = \mathbf{X}(y_t, ..., y_{t-d+1}) \cdot \mathbf{C} \tag{5}$$

In this work the methods and models were implemented in the Python programming language[36–38] and the aim was to forecast a total of 24 steps ahead, $H = 24$, using the same number of input values, $d = 24$. The full set of data consisted on a total of 8784 data points from which 80% were used for training and cross validation and the remaining 20% for testing as suggested by Pareto's principle.[39] From the 6980 data points used for training a ten-fold winner takes all cross validation process was used. This was done by splitting that portion of the data into ten equal parts, use nine of them to train the model and the remaining one to validate it. This process was repeated ten times always changing the validation segment. Afterwards, the iteration that yielded the best results was replicated and tested on the remaining 1804 data points that comprised the test set. The testing process was done three times on the same set but with different start and ending points to account for variations of the model and to add statistical meaning to the results.[26]

The performance of the model was evaluated using the mean average percentage error (MAPE) which is the standardized approach to demonstrate the prediction accuracy of a forecasting method in statistics and machine learning theory.[28,40] It is given by:

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \tag{6}$$

where A_t is the actual value and F_t is the corresponding forecasted value for a data set of size n. The multiplication by 100 is necessary to convert and present the value as a percentage.

Another aspect explored was that of seasonality extraction or seasonal decomposition and its effects on model performance.[26] This was done by smoothing the original data set with the Savitzky-Golay filer generating an extra data set which was submitted to the same

testing process.

Results and discussion

Throughout this section all time series set, both original and deseasonalized, will be empirically considered as stationary based on the prior application of the Dickey-Fuller test. The focus is on analyzing and interpreting the results of the MLA application on all time series. Outcomes from the application on the undecomposed time series are the primary focus and serve as a term of comparison for further results involving decomposed time series.

Original data

The results of the MLA application on the original data without any kind of seasonality extraction are shown in the plots of figure 2. Starting from sub-figures 2a and 2b, which pertain to the results of PM10 forecasting, the dispersion spread of the intersects between real and forecasted values suggests a good model since most data points intersect very close, and even on top, the quadrant bisector line. This assessment seems to be further supported by the plot in sub-figure 2b which shows the forecasted values (red line) alongside the real values (blue line) on the same time scale. The two lines are very close together and individual errors, represented bellow by the green line, are mostly very small only except in a few situations which correspond to inflection points of the original signal. In addition, the mean MAPE value for all three testing iterations is of 27.53 % and the dashed line in the lower error plot.

For PM2.5 forecasting a similar case is presented in sub-figures 2c and 2d, which was expected since bot PM2.5 and PM10 are highly correlated variables which share the same sources. The spread of the intersects is slightly worst than that of PM10 forecasting but it is still very acceptable and indicates a similar degree of consistency. the error plot in sub-figure 2c also shows that the forecasted values are in good proximity with the real values along the same time axis and individual error measurements are mostly all below the MAPE

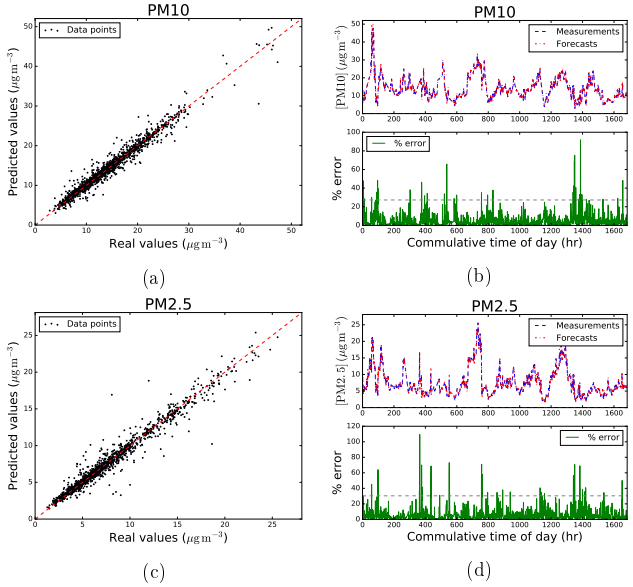


Figure 2: Dispersion and error plots for the third test iteration of PM10 and PM2.5 forecasting without seasonality decomposition. a) PM10 3^{rd} test iteration dispersion plot; b) PM10 3^{rd} test iteration error plot; c) PM2.5 3^{rd} test iteration dispersion dispersion plot; d) PM2.5 3^{rd} test iteration error plot

value with a few exceptions, which, like in PM10 forecasting, correspond to the inflection points of the time series. The mean MAPE value is of 30.63 % which is not as low as the PM10 forecasting MAPE, of 27.53 %, but it is still a good vale.

Seasonal decomposed data

In figure 3 are presented the plots with the results from the application of the MLA on the filtered data (using a third degree polynomial Savitzky-Golay filter). Sub-figures 3a and 3b show the PM10 forecasting results and on the left-sided plot it is noticeable that there exists great prediction coherence given that the correlation between real and forecasted values is very high given the overlap of the data points with the quadrant bisection. In the other

plot, the upper part shows that the real (blue line) and forecasted (red line) values have the same behavior and are very close with each other while in the lower plot individual errors are shown to remain relatively constant with almost none being higher than the MAPE value. The mean MAPE value is indicated by the dashed line in the lower right-sized plot and is of 26.451 %.

The case for PM2.5 forecasting is similar considering the use of the Savitzky-Golay filter. The analysis of sub-figure 3c yields the same conclusions with most data points overlapping the quadrant bisector line suggesting that great correlation and coherence exists between real and forecasted values. In addition, the upper plot presented in sub-figure 3d shows that behavior of the forecasted values and real values is almost identical and the lower plot in the same sub-figure shows relatively constant individual errors with only a few exceptions that seem to correspond to infection points in the original signal. The mean MAPE is of 29.420 %. Analogous to the original signal based MLA, this value is not as low as its PM10 counterpart, but is still very decent.

Comparative analysis

In table 1 a summarized version of the results obtained by the application of the MLA both data sets, filtered and unfiltered, can be found. In this table also presented the standard deviation generated while averaging every iteration MAPE value. These values are of great importance because they can be used as a tie-breaker criteria for choosing the best model outcome.

Beginning with PM10 forecasting, the application of the MLA on the original and the filtered data sets yielded similar results with the later out-performing the original by a slight margin. Model performance obtained by using the original signal is evaluated by the mean MAPE value of 27.539 % while the same value using the filtered signal was of 26.451 %. This improvement occurred not only in terms of accuracy but also regarding forecast coherence given that the filtered signal standard deviation was smaller than that of the original signal

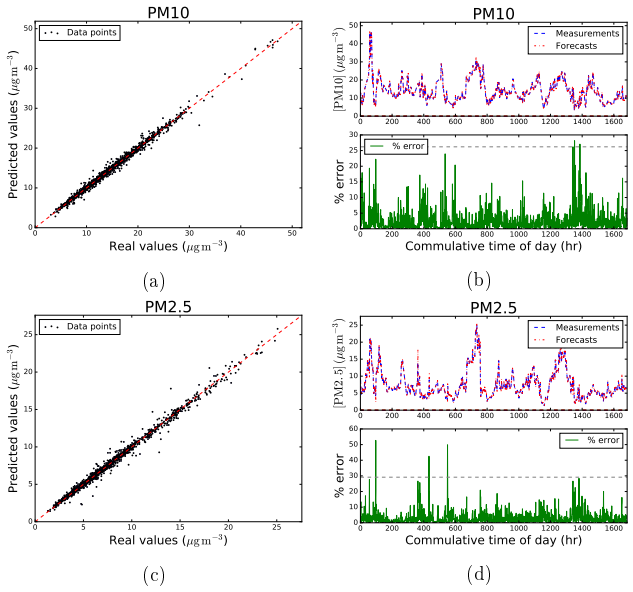


Figure 3: Dispersion and error plots for the third test iteration of PM10 and PM2.5 forecasting with the application of the savitzky-Golay filter. a) PM10 3^{rd} test iteration dispersion plot; b) PM10 3^{rd} test iteration error plot; c) PM2.5 3^{rd} test iteration dispersion dispersion plot; d) PM2.5 3^{rd} test iteration error plot

Table 1: Summary of forecasting model results. Note: all values except for standard deviations (Stdev) are expressed in percentages.

MAPE		Original	Savitzky-Golay
PM10	**Test 1**	27.667	26.563
	Test 2	27.670	26.591
	Test 3	27.281	26.200
	Mean	27.539	**26.451**
	Stdev	0.224	**0.218**
PM2.5	**Test 1**	30.880	29.650
	Test 2	30.649	29.443
	Test 3	30.369	29.164
	Mean	30.632	**29.420**
	Stdev	0.256	**0.244**

with the values of 0.218 and 0.224 respectively.

PM2.5 forecasting results are very similar, in interpretation, to those of PM10 forecasting. The application of the MLA on the original signal yielded a mean MAPE value of 30.632 % with a standard deviation of 0.256 while the same application on the filtered signal yielded values of 29.420 % and 0.244 for the mean MAPE and standard deviation respectively. This implies using the Savitzky-Golay filter seemed to improve both accuracy and coherence by a small margin also for PM2.5 forecasting, however the original signal also generated a usable model.

Conclusion

Multi-step ahead time series forecasting is a very hard task and severely limited by the uncertainty of predictions for large forecasting horizons. Linear regression machine learning algorithms are among the most widely used to address these problems given that they are very versatile and easy to implement. Substantiated by the knowledge that approaches based on multivariate regression have been shown to be regularly better than those based of multiple regression an attempt at forecasting PM10 and PM2.5 variations in the atmosphere

using a multi-output MLA was the end goal for this project.

The developed MLA featured in this work was able to predict PM10 and PM2.5 variations coherently and replication of results was easily achieved to account for proper statistical meaning. Although performance metrics are bellow initial expectations the final model can be safely used in a real life situation even though it could benefit from further improvements.

A comparative analysis regarding the seasonal decomposition of the data was also done and the main conclusion is that applying a Savitzky-Golay filter prior to traininga nd testing the MLA managed to improve its performance slightly both in terms of accuracy and coherence for PM10 and PM2.5 forecasting.

Overall, the development of a usable MLA was achieved but further improvements can be made by training and testing using bigger and more variable data sets to account for variations and fluctuations which were not present in the original data. Also, the incorporation of a non-linear function could almost certainly improve forecasting accuracy by compromising the low usage of computational time and resources. Lastly, finding a way to introduce variable correlation, via blind source separation techniques, could also improve the present model.

Acknowledgement

The authors thank the Coimbra Chemistry Centre (CQC) who is supported by the Portuguese "Fundação para a Ciência e a Tecnologia" (FCT), through the project QUI/UI0313/2013, co-funded by COMPETE-UE and SpaceLayer Technologies who participates in the European Space Agency Business Incubation Center Portugal (ESA-BIC Portugal) and the Copernicus accelerator.

References

(1) European Environment Agency, *State and Outlook 2015 the European Environment*;

2015.

(2) World Health Organization, Ambient (outdoor) air quality and health. 2016.

(3) Wang, L.; Zhong, B.; Vardoulakis, S.; Zhang, F.; Pilot, E.; Li, Y. *Int. J. Environ. Res. Public. Health.* **2016**, *13*, 1196–1220.

(4) Chu, Y.; Liu, Y.; Li, X.; Liu, Z.; Lu, H.; Lu, Y.; Mao, Z.; Chen, X.; Li, N.; Ren, M.; Liu, F.; Tian, L.; Zhu, Z.; Xiang, H. *Atmosphere* **2016**, *7*, 129–154.

(5) Katsouyanni, K. *Brit. Med. Bul.* **2003**, *68*, 143–156.

(6) Macintyre, H.; Heaviside, C.; Neal, L.; Agnew, P.; Thornes, J.; Vardoulakis, S. *Environ. Int.* **2016**, *97*, 108–116.

(7) Xie, R.; Sabel, C.; Lu, X.; Zhu, W.; Kan, H.; Nielsen, C.; Wang, H. *Environ. Int.* **2016**, *97*, 180–186.

(8) Taheri Shahraiyni, H.; Sodoudi, S. *Atmosphere* **2016**, *7*, 15.

(9) Matti Maricq, M. *J. Aerosol Sci.* **2007**, *38*, 1079–1118.

(10) Amato, F.; Pandolfi, M.; Moreno, T.; Furger, M.; Pey, J.; Alastuey, A.; Bukowiecki, N.; Prevot, A.; Baltensperger, U.; Querol, X. *Atmos. Environ.* **2011**, *45*, 6777–6787.

(11) Lenschow, P. *Atmos. Environ.* **2001**, *35*, 23–33.

(12) Keary, J.; Jennings, S.; O'Connor, T.; McManus, B.; Lee, M. *Urban Air Quality: Monitoring and Modelling*; Springer Netherlands: Dordrecht, 1998; pp 3–18.

(13) de Mattos Neto, P.; Cavalcanti, G.; Madeiro, F.; Ferreira, T. *PLoS ONE* **2015**, *10*, 138–507.

(14) Gautam, S.; Yadav, A.; Tsai, C.; Kumar, P. *Environ. Sci. Pollut. R.* **2016**, *23*, 21165–21175.

(15) Fajersztajn, L.; Veras, M.; Barrozo, L.; Saldiva, P. *Nat. Rev. Cancer* **2013**, *13*, 674–678.

(16) Feng, J.; Yang, W. *PLoS ONE* **2012**, *7*, e33385.

(17) Kleinman, M. *Central nervous system effects of ambient particulate matter: the role of oxidative stress and inflammation final report*; 2014.

(18) Breitner, S.; Schneider, A.; Devlin, R.; Ward-caviness, C.; Diaz-sanchez, D.; Neas, L.; Cascio, W.; Peters, A.; Hauser, E.; Shah, S.; Kraus, W. *Environ. Int.* **2016**, *97*, 76–84.

(19) Gozzi, F.; Della Ventura, G.; Marcelli, A. *Atmospheric Pollution Research* **2016**, *7*, 228–234.

(20) Spurnyy, K. *Advances in aerosol filtration*; Lewis Publishers, 1998; p 533.

(21) Harrison, R.; Deacon, A.; Jones, M.; Appleby, R. *Atmospheric Environment* **1997**, *31*, 4103–4117.

(22) Querol, X.; Alastuey, A.; Ruiz, C.; Artinãno, B.; Hansson, H. .; Harrison, R.; Buringh, E.; Ten Brink, H.; Lutz, M.; Bruckmann, P.; Straehl, P.; Schneider, J. *Atmospheric Environment* **2004**, *38*, 6547–6555.

(23) Wilks, D. *Statistical methods in the atmospheric sciences*; Academic Press, 2011; p 676.

(24) Lynch, P. *Journal of Computational Physics* **2007**, *227*, 3431–3444.

(25) Nazif, A.; Mohammed, N.; Malakahmad, A.; Abualqumboz, M. *Water, Air, and Soil Pollution* **2016**, *227*, 1–12.

(26) Ben Taieb, S.; Bontempi, G.; Atiya, A. F.; Sorjamaa, A. *Expert Syst. Appl.* **2012**, *39*, 7067–7083.

(27) Pérez, P.; Trier, A.; Reyes, J. *Atmospheric Environment* **2000**, *34*, 1189–1196.

(28) Sfetsos, A. *J. Software Eng. Appl.* **2010**, *03*, 374–383.

(29) SpaceLayer-Technologies, *"Go-to Market Plan", Building Global Innovators, MIT Portugal*; 2015.

(30) European Environment Agency, Atmosphere Monitoring | Copernicus. 2014; `http://copernicus.eu/main/atmosphere-monitoring;Accessed:2017-01-30`.

(31) Dickey, D.; Fuller, W. *J. Am. Stat. Assoc.* **1979**, *74*, 427.

(32) R Spiegel, M.; J. Stephens, L. In *Theory and Problems of Statistics*, 3rd ed.; Barbara Gilson„ Ed.; McGraw-Hill: New York, 1999; p 2016.

(33) Bontempi, G. *Proc. 2nd ESTSP* **2008**, 145–154.

(34) Ben Taieb, S.; Sorjamaa, A.; Bontempi, G. *Neurocomputing* **2010**, *73*, 1950–1957.

(35) Björck, Å. *Numerical Methods for Least Squares Problems*; Society for Industrial and Applied Mathematics, 1996.

(36) Harrington, P. *Machine Learning in Action*, 1st ed.; Manning Publications: New York, 2010; p 382.

(37) Van Rossum, G.; Drake, F. *The Python Language Reference Release 3.2.3*; 2012; p 121.

(38) Pedregosa, F. et al. *Journal of Machine Learning Research* **2012**, *12*, 2825–2830.

(39) Kiremire, A. *The application of the Pareto Principle in software engineering*; 2011; Vol. 13.

(40) Makridakis, S. *International Journal of Forecasting* **1993**, *9*, 527–529.