Deep learning based prediction and visual analytics for temporal environmental data

Von der Fakultät Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Shubhi Harbola

aus Nainital, India

Hauptberichter:	Prof. Dr. Volker Coors
Mitberichter:	Prof. Dr. Thomas Ertl
	Prof. (em.). Robert Laurini
Tag der mündlichen Prüfung:	07. Dec. 2021

Institut für Visualisierung und Interaktive Systeme
der Universität Stuttgart

2022

Contents

List of Figures

List of Tables

Acknowledgments

I would like to extend my sincerest thanks to my supervisor Prof. Volker Coors, who gave me the opportunity to become a PhD student under Joint Graduate Research Training Group Windy Cities (Windy Cities) and gave me guidance and support throughout this entire time. Further, I would like to thank my co-supervisor, Prof. Thomas Ertl, for agreeing to supervise my thesis and for his constant and dedicated support throughout my PhD work. They allowed me to think independently and differently and gave me the freedom to work in my own ways. I also thank Prof. (em.). Robert Laurini for accepting to serve as an external examiner for my thesis.

I am also grateful to all the anonymous reviewers (of the manuscripts that I have submitted in different venues) for their suggestions and comments, which helped improve my writing skills and understanding of the current research.

I am thankful to Windy Cities for funding, interdisciplinary learning, valuable insights into the simulation and visualisation development domains for wind analyses, and my friends and colleagues in Windy Cities.

I extend my sincerest gratitude to Dennis Thom for giving me some of the best earlier stage research wisdom and Steffen Koch for the valuable discussions about my work and support. They were available to guide me and boosted my confidence when required.

In addition, I would like to thank Anja Ernst for helping me with most of the official formalities and supporting me on my initial days of settlement in Stuttgart.

I thank all my officemates and colleagues at HFT and VIS for being kind and supportive. Perhaps I have received the best of both University of Stuttgart and HFT Stuttgart, making it possible to bring this work to a successful end. I feel indebted and will always remain grateful.

I would like to express my gratitude to my parents for their unlimited support, love and help. Special thanks to bhai for being willing to help and give his best suggestions.

Abstract

Environmental analyses require detailed understanding and supporting analytics of the surroundings together with visualisation for easy interpretation. Many cities are now providing open environmental data, but the online analysis capabilities in their open data platforms are usually weak or non-existent. Moreover, increasing the efficiency of maintaining and planning the detailed activities require an increase in the amount of digitally available environmental and surrounding monitoring awareness. This may need that the environmental data is collected continuously through sensors. The relationships of environmental data (meteorological and pollution parameters) and their variations make prediction estimations of how their distributions vary in space and time very important. The demand for a reliable prediction algorithm that would work directly on the original historical temporal environmental data, without any transformation, on a large dataset and with the user-defined time frame of prediction in future with adequate accuracy (close to actual or reality) is still challenging. Moreover, a technique is required that can help in automated analyses with interactive visualisation, thereby assisting in the easier understanding of the spatio-temporal environmental data along with decision making capabilities. A framework that combines the above prediction and interactive visualisation in a single platform is also desirable.

The prediction can be achieved using Machine Learning models comprising deep learning algorithms. These Machine Learning techniques including deep learning are subsets of Artificial Intelligence. The inheriting qualities to self learn the insightful patterns and trends directly from the data makes deep learning a comprehensive and favourable methodology in order to automate seasonality and information extraction for the environment domain. Visual Analytics increases the value by combining machines' processing power and accuracy with the human capabilities to perceive information visually, fuse, and aggregate the data and detect hidden patterns therein. The integrated prediction and visualisation framework in a web based platform would increase the trust in data, models, and results, which is especially important when decisions are needed to be based on environmental analyses and quality assessments. These aspects provide the

requisite research motivation of designing the meteorological and pollution parameters visual prediction temporal analyses model.

In view of the above, the objective of this thesis is to focus on developing Machine Learning methods and their visualisation for environmental data. The presented approaches primarily focus on devising an accurate Machine Learning framework that supports the user in understanding and comparing the model accuracy in relation to essential aspects of the respective parameter selection, trends, time frame, and correlating together with considered meteorological and pollution parameters. Later, this thesis develops approaches for the interactive visualisation of environmental data that are wrapped over the time series prediction as an application. Moreover, these approaches provide an interactive application that supports

1. a Visual Analytics platform to interact with the sensors data and enhance the representation of the environmental data visually by identifying patterns that mostly go unnoticed in large temporal datasets,

2. a seasonality deduction platform presenting analyses of the results that clearly demonstrate the relationship between these parameters in a combined temporal activities frame, and

3. air quality analyses that successfully discovers spatio-temporal relationships among complex air quality data interactively in different time frames by harnessing the user's knowledge of factors influencing the past, present, and future behaviour with Machine Learning models' aid.

Some of the above pieces of work contribute to the field of Explainable Artificial Intelligence which is an area concerned with the development of methods that help understand, explain and interpret Machine Learning algorithms. In summary, this thesis describes Machine Learning prediction algorithms together with several visualisation approaches for visually analysing the temporal relationships among complex environmental data in different time frames interactively in a robust web platform. The developed interactive visualisation system for environmental data assimilates visual prediction, sensors' spatial locations, measurements of the parameters, detailed patterns analyses, and change in conditions over time. This provides a new combined approach to the existing visual analytics research. The algorithms developed in this thesis can be used to infer spatio-temporal environmental data, enabling the interactive exploration processes, thus helping manage the cities smartly.

Zusammenfassung

Umweltanalysen erfordern ein genaues Verständnis sowie unterstützende Analysen der Umgebung in Kombination mit Visualisierung, um Analyseergebnisse leichter interpretierbar zu machen. Viele Städte stellen inzwischen Umweltdaten offen zur Verfügung, allerdings sind Online-Analysefunktionen in deren offenen Datenplattformen in der Regel nur rudimentär oder gar nicht vorhanden. Darüber hinaus erfordert eine Effizienzsteigerung bei der Wartung und Planung detaillierter Aktivitäten eine wesentlich größere Menge an digital verfügbaren Umwelt- und Umgebungsdaten. Dies macht es mitunter erforderlich, Umweltdaten mit Hilfe von Sensoren kontinuierlich zu sammeln. Aufgrund der Abhängigkeiten zwischen Umweltdaten (z.B. zwischen meteorologischen Daten und Verschmutzungsdaten) sowie deren Schwankungen, sind Vorhersagen darüber, wie sich deren Verteilungen in Raum und Zeit ändert, von großer Bedeutung. Die Forderung nach einem zuverlässigen Vorhersagealgorithmus, der direkt mit den ursprünglichen zeitlich dynamischen Umweltdaten arbeitet, ohne auf die Transformation eines großen historischen Datensatzes zurückzugreifen, und mit dem, in einem von Benutzerinnen und Benutzern zuvor festgelegten Zeitraum, eine Vorhersage mit angemessener Genauigkeit (nahe an der Realität) realisiert werden kann, ist immer noch eine Herausforderung. Darüber hinaus wird eine Technik benötigt, die automatisierte Analysen mit interaktiver Visualisierung kombiniert und damit zum Verständnis räumlich-zeitlicher Umweltdaten beiträgt und die Entscheidungsfindung unterstützt. Ein Framework, das die oben genannte Vorhersage und interaktive Visualisierung in einer einzigen Plattform kombiniert, ist ebenfalls wünschenswert.

Die Vorhersage kann mithilfe von Modellen des maschinellen Lernens erfolgen, die nicht zuletzt auf Deep Learning-Algorithmen basieren. Techniken des maschinellen Lernens, einschließlich Deep Learning, sind Teilgebiete der Forschung zu Künstlicher Intelligenz (Artificial Intelligence). Die inhärente Fähigkeit, aufschlussreiche Muster und Trends direkt aus den Daten zu lernen, macht Deep Learning zu einer geeigneten Methode, die Erkennung wiederkehrender Effekte in den Daten und Informationsextraktionen für den Umweltbereich zu automatisieren. Visual Analytics kann

dabei einen Mehrwert schaffen, indem es die Verarbeitungsgeschwindigkeit und -genauigkeit von Computern mit den menschlichen Fähigkeiten kombiniert, Informationen visuell wahrzunehmen, Daten zu fusionieren und zu aggregieren, sowie darin verborgene Muster zu erkennen. Ein webbasiertes Framework für die integrierte Vorhersage und Visualisierung könnte das Vertrauen in Daten, Modelle und Ergebnisse erhöhen. Dies ist besonders wichtig, wenn Entscheidungen basierend auf Umweltanalysen und Qualitätsbeurteilungen getroffen werden müssen. Diese Überlegungen bilden die Forschungsmotivation für die Entwicklung eines Analysemodells zur visuellen Vorhersage zeitabhängiger meteorologischer Daten sowie von Verschmutzungswerten.

Vor diesem Hintergrund ist das Ziel dieser Arbeit die Entwicklung von Methoden des maschinellen Lernens und deren Visualisierung für Umweltdaten. Die vorgestellten Ansätze konzentrieren sich in erster Linie auf die Entwicklung eines präzisen Frameworks für maschinelles Lernen, das Nutzerinnen und Nutzer beim Verständnis und Vergleich der Modellgenauigkeit in Bezug auf wesentliche Aspekte der Parameterauswahl, Trends, Zeitrahmen und Korrelationen der betrachteten meteorologischen Daten und Schadstoffparameter unterstützt. Im weiteren Verlauf dieser Arbeit werden Ansätze zur interaktiven Visualisierung von Umweltdaten entwickelt, welche Vorhersagen über Zeitreihen in eine entsprechende Anwendung integrieren. Darüber hinaus ermöglichen diese Ansätze eine interaktive Anwendung, die folgende Optionen bietet:

1. Eine Visual-Analytics-Plattform zur Interaktion mit den Sensordaten und zur Verbesserung der visuellen Darstellung von Umweltdaten durch Identifizierung von Mustern, die sonst in großen zeitlichen Datensätzen meist unbemerkt bleiben,

2. eine Webplattform zur Erkennung zeitlich wiederkehrender Effekte, die Ergebnisanalysen präsentiert, um Beziehungen zwischen Parametern in einer kombinierten Zeit-/Aktivitätsspanne deutlich machen, und

3. Luftqualitätsanalysen, die räumlich-zeitliche Beziehungen zwischen komplexen Luftqualitätsdaten interaktiv in verschiedenen Zeitrahmen erfolgreich aufdecken, indem sie auf das Nutzerwissen über vergangene, gegenwärtige und zukünftige Einflussfaktoren mit Hilfe von Modellen des maschinellen Lernens zurückgreifen.

Einige der oben genannten Arbeiten leisten einen Beitrag zum Forschungsfelds Explainable Artificial Intelligence, einem Fachgebiet, das sich mit der Entwicklung von Methoden befasst, die helfen, Algorithmen des

maschinellen Lernens besser zu verstehen, zu erklären und zu interpretieren. Zusammengefasst werden in dieser Arbeit maschinelle Lernverfahren für die Vorhersage zusammen mit verschiedenen Visualisierungsansätzen zur visuellen, interaktiven Analyse der zeitlichen Beziehungen zwischen komplexen Umweltdaten in verschiedenen Zeiträumen als Teile einer robusten webbasierten Anwendung vorgestellt. Das entwickelte interaktive Visualisierungssystem für Umweltdaten integriert die visuelle Vorhersage, räumliche Positionierung von Sensoren, Messungen der Parameter, detaillierte Musteranalysen und zeitliche Änderungen der Situation. Dies erweitert die bisherige Forschung im Bereich Visual Analytics um einen neuen kombinierten Ansatz. Die in dieser Arbeit vorgestellten Algorithmen können verwendet werden, um die räumlich-zeitliche Entwicklung von Umweltdaten abzuschätzen. Damit ermöglichen sie interaktive Erkundungsprozesse und tragen so zu einem intelligenten Stadtmanagement bei.

1

Introduction

The environmental data focusing on meteorological and pollution parameters, is an important aspect for monitoring environmental conditions, ambient air quality, efficient management and understanding of renewable resources (Biber, 2013). Correlating and combining the meteorological information with pollution parameters facilitate in environmental data understanding. The meteorological data comprises parameters like pressure, temperature, wind and humidity. The pollution parameters combine city air pollutants such as Particular Matter (PM_{10}, $PM_{2.5}$), Nitrogen Oxide (NO), Nitrogen Dioxide (NO_2), and Ozone (O_3). The large volume of temporal environmental data is continuously collected and monitored throughout the cities, including spatial information based on the sensor's position. It is difficult to analyse and understand the insights of this acquired raw environmental data without a proper framework. Moreover, complex and volatile environmental data are challenging to analyse in different time frames along with estimating their nature for the future confidently. This has increased the demand for an intelligent framework for analyses and visualisation using Machine Learning (ML) integrated with Visual Analytics (VA) concepts for environmental data. This would provide the ability to transform the collected environmental raw data into meaningful information for better decision making. ML aids in the detailed pattern analyses and prediction of the diverse environmental data having the aforementioned characteristics. VA provides faster decisions making, as users can understand data insights much more quickly by seeing and working with datasets when they are in a visual format. Moreover, providing detailed analyses supported by ML advanced techniques along with delivering the findings and insights by VA convenience is a required solution for these volatile temporal parameters and air quality conditions.

Section 1.1 explains ML concepts briefly (related concepts are explained in section 2.4, followed by the thesis ML work in Chapter 3), section 1.2

gives an overview of VA (related concepts are expanded in section 2.2, followed by the thesis VA related work in Chapter 4 and Chapter 5), section 1.3 explores the current trends in the time series visual prediction for environmental data, and section 1.4 and section 1.5 highlight the issues in the existing literature and the thesis objective, respectively, followed by section 1.6 that discusses the structure and contribution of the thesis.

1.1 Machine Learning

ML provides algorithms that learn about the given dataset by using training samples, and after the algorithms have been trained, testing is done on samples that are not part of training samples. ML has opened up new possibilities and approaches for applications in environmental data (Cho, 2018; Lamba et al., 2019).

Presently the existing models for environmental data temporal prediction (discussed in detail in section 2.10) have used limited datasets to analyse the models and predict only a few values in the future (Liu et al., 2018). However, a prediction model that would work directly on the original temporal dataset, without any transformation, and over a large historical dataset for the user-defined time frame of prediction in the future is still required.

1.2 Visual Analytics

VA is a sub field of visualisation which integrates data analyses with interactive visualisations (Thomas and Cook, 2005; Isenberg et al., 2017). The visualisation of spatio-temporal data in an interactive temporal time frame is essential for VA of environmental data, magnifying the insight of data in a visual context by identifying trends. These patterns usually go unrecognised in voluminous historical temporal environmental data. This helps in developing techniques representing spatio-temporal data in more sophisticated formats using maps, detailed bars, charts, and heat maps to communicate the relationships between the sensor measurements. Furthermore, VA combines automated analyses techniques with interactive visualisation, thereby assisting in the easier understanding of the spatio-temporal data along with decision making capabilities. Thus, there is a need for research on effective solutions for detail analyses of the environmental data, taking into account the resources that are available.

1.3 Time Series Visual Prediction

The visual exploration of past, present and future trends in complex multivariate time series environmental datasets plays an important role in better judging the environmental conditions by finding relationships between meteorological and pollution parameters. Including the context and historical information in the visualisation could improve user understanding of the environmental dataset exploration process and enhances the reusability of mining and managing techniques and parameters analysis to achieve the required insights.

The traditional approaches (more details in section 2.11) cannot fully support the visual exploration of future trends in complex multivariate time series datasets such as environmental data, mainly due to their lack of consideration of inter-variable parameters relationships (Hao et al., 2011; Köthur et al., 2012). A platform is still required to support the user in formulating hypotheses about the environmental data that may be useful for further stages of the mining process, such as cluster detection, important feature and pattern detection, with interactive visualisation options. Furthermore, the platform could give a provision of including the prediction of meteorological and pollution parameters in the desired time frame with the especially designed deep learning models support, along with highlighting the respective model's success and failure, suggesting the best option to choose. Additionally, it could provide the freedom to the users to compare and analyse the environmental data as per their selection in the considered time frame. Moreover, it would justify the arguments with easy graphical support, along with historical, present, and future data patterns that could be visualised under one platform.

1.4 Problem

Many cities are providing open environmental data, but the online analysis capabilities in their open data platforms are usually weak or non-existent. The research that could be used to fill this gap is a highly relevant and desirable development. The literature survey of the above topics in chapter 2 reveals that an approach is still required that can combine ML prediction algorithms together with VA techniques for visually analysing the temporal relationships among environmental data in different time frames interactively in a web platform. This motivates the below research objective that is answered in this thesis.

1.5 Thesis Objective

The main objective of this thesis is to visually analyse the temporal relationships interactively in different time frames (past, future) for the environmental data. The main difference between the developed techniques and state-of-the-art (discussed in detail chapter 2) is providing a solution combining the advantages of both ML and visualisation for temporal environmental data. The primary focus is to devise an accurate deep learning framework support at the backend with an interactive visualisation frontend to perform the prediction analyses and environmental data interaction. This objective is further divided into the following pieces of work:

1. Use ML including deep learning algorithms for environmental data predictions in the future.

2. As an expanded application for these predictions, visualise and evaluate these predictions' results and algorithms' assessments using visual analytics concepts for time series data.

3. Development of an interactive web platform that combines these predictions and analyses for selected time frame and meteorological and pollution parameters.

The above objective and some sub tasks also contribute to the field of Explainable Artificial Intelligence (XAI), which is an area that is concerned with the development of new methods that explain and interpret ML algorithms (Choo and Liu, 2018; Xie et al., 2020). Furthermore, the above objective

1. helps to understand the temporal relations between meteorological and pollution parameters,

2. explains prediction outputs of ML through VA for environmental data,

3. highlights the respective sensor's location along with measurements whenever a query related to the sensor nature monitoring is performed. Here a comparison is performed among environmental sensors at different spatial locations. Finally, the sensor location is predicted that would measure the highest value of the selected parameter for the predicted time frame.

1.6 Structure and Contributions

This section outlines the rest of this thesis and gives an overview of each chapter's content. It is divided into sub tasks (section 1.5) that are part of the (refer chapter 6.2) published papers and form each individual chapter, highlighting the findings. Each chapter of this thesis acts as a building block that is improved, integrated and advanced to achieve the desired solutions. I am the first author of all the publications that are presented in these chapters. These findings and developments have passed double blind peer review during a publication process already (refer chapter 6.2).

Chapter 2 - Fundamentals & Background: This chapter introduces the concepts and techniques that are relevant during the remainder of the thesis. The first part presents the general concept of GeoVisualisation and VA, and the necessary visualisation fundamentals. This chapter is based on the published survey paper Harbola and Coors (2018). The second part of the chapter provides a comprehensive overview of the development from traditional to advanced ML based prediction approaches, data pre-processing, and analysis algorithms implemented for environment data (meteorological and pollution parameters). Furthermore, discussing the visualisation technique for interactive prediction of time series data helps in understanding essential visualisation concepts. Moreover, ML and VA integration importance is highlighted along with the role of environmental analyses combined together with ML and VA using methods which are empirical ML architectures and wrapped with the advantages of interactive visualisation to help plan our surroundings.

Chapter 3 - Machine Learning Algorithms for Predictions: This chapter introduces techniques that are designed for environmental data temporal prediction. This contributes to the first thesis objective task of implementing future prediction of the environmental data. In this thesis, five advanced ML and deep learning temporal prediction models to analyse the environmental data, have been developed. This work is part of the following papers Harbola and Coors (2019a,b,c, 2021a). These algorithms take successive time values in terms of environmental data as input and predict the future nature. The advantage of these methods is that they do not apply any smoothening and noise removal techniques and are based on a classification approach. Each designed model is the advancement and extension of the previous method (lacking part is improved in the next model design) in terms of architectures, accuracy, classes, design, and multiple features. A total of 58 features in the input layers are designed

by employing recent ML concepts. The multiple features are based on percentage difference, standard deviation, correlation coefficient, eigenvalues, and entropy for efficiently describing and exploring the data trend. The sections of the designed techniques are summarised as follows:

Long Short Term Memory, Random Forest & Support Vector Machine: (section 3.1) The first technique is based on the temporal prediction of the wind flow, using three supervised algorithms $i.e.$, Long Short Term Memory (LSTM), Random Forest (RF) and Support Vector Machine (SVM). These algorithms take successive time values in terms of wind speed and direction as input and predict the future dominant wind flow, as classification approach. The developed algorithms are trained and tested using historical wind datasets of Stuttgart (Germany) and Netherlands. The total accuracy of prediction using LSTM and SVM were similar and reached up to 94.7%, providing an improvement over RF. The advantage of these methods is that they do not apply any smoothening and noise removal techniques and are based on classification approach. LSTM learns long term dependencies in the temporal data, SVM finds the probable hyperplane between points of different classes and RF uses multiple decision trees. However, in these algorithms a limited number of features and classes are used. A better approach is required that could incorporate multiple features and more number of classes.

One-Dimensional Convolutional Neural Network Architectures: (section 3.2) The second technique is based on One-Dimensional (1D) Convolutional Neural Network (CNN). The developed 1D Single CNN (1DS) takes as input the temporal values in terms of the wind speed and direction. The 1DS comprises several convolutional layers along with fully connected layers, that learn automatically numerous spatial and non-spatial features at different scales during the training process. The developed third technique is 1D Multiple CNN (1DM) that combines several 1DS but with different views of the same input, therefore, learning more information compared to the 1DS. The 1DS and 1DM algorithms are trained and tested for Stuttgart and Netherlands datasets with the achieved total accuracies of 95.2% (1DS) and 99.7% (1DM). The 1DS improves upon LSTM, RF, and SVM methods by using more number of classes (eleven) and higher number of automatically learnt features in the convolutional and fully connected layers, that enhances the accuracy. Further, the 1DM has better performance than the 1DS due to the use of multiple 1DS. In these approaches limited (only two) features based on wind speed and direction are used in the input layers. Further, the fully connected layers do not have memory to retain the features learnt by neurons from the previous training iterations. Thus, an algorithm is desired that could take multiple features in the input layers as well.

Multiple Densely Connected Convolutional Neural Network: (section 3.3) The fourth technique builds upon the 1DM method and proposes a multiple CNN architecture with multiple input features, combined with multiple LSTM, along with densely connected convolutional layers. The designed architecture is called Multiple features, Multiple Densely Connected Convolutional Neural Network ensembles with Multiple LSTM Architecture $i.e.$, MCLT. A total of 58 multiple features in the MCLT input layers are designed using wind flow values. These empirical features are based on percentage difference, standard deviation, correlation coefficient, eigenvalues, and entropy, for efficiently describing the wind trend. Two successive LSTM layers are used after four densely connected convolutional layers of the MCLT. LSTM has memory units that utilise learnt features from the current as well as previous outputs of the neurons, thereby enhancing the learning of patterns in the temporal wind dataset. The presence of densely connected convolutional layers help to learn features of other convolutional layers as well. The MCLT uses 21 classes for prediction unlike eleven classes in the 1DM and performs better. The maximum total accuracy is 99.9%. However, in the above discussed ML based methods for prediction, there is a lack of visualisation as required in VA. Thus, an approach is required that helps in the visualisation of different patterns in the dataset for different time frames.

Chapter 4 - Seasonality Deduction Application: This chapter presents the ML models' application in performing the VA. The second objective of this work is expanded to help in understanding the temporal relation between meteorological and pollution parameters interactively. Specifically, it first presents an interactive dashboard to visualise meteorological and pollution parameters for the desired time frame. This helps to analyse the case study area's temporal variations. Furthermore, the correlations between meteorological and pollution parameters are analysed with the help of this technique. Some of these findings are part of the following papers Harbola and Coors (2020); Harbola et al. (2021b). In the fifth technique, emphasis is on the seasonality deduction for the pollution parameters in relationship with the meteorological parameters. However, an improved approach is required that combines more environmental data, correlation analysis, temporal heat map and a better interactive visualisation integrating with the above developed ML visual predictors for multiple parameters in depth analysis for various time frames in a robust web platform.

Chapter 5 - Air Quality Temporal Analyser & Geospatial Data Visual Assessments: This chapter discusses the third thesis objective to ex-

plain prediction outputs of ML through VA for environmental data and supports analysing the nature of a sensor for the selected meteorological and pollution parameter. The approaches presented in this chapter focus on supporting the application of these above designed ML models into VA. This chapter presents visualisation and VA interface for the spatio-temporal data represented in terms of sensors's location, including time, and several environment attributes to assess the detailed temporal patterns of these parameters for combined interactive platform analyses. These work findings are part of the following papers Harbola et al. (2021a); Harbola and Coors (2021b).

Air Quality Temporal Analyser: (section 5.1) The developed technique is an Air Quality Temporal Analyser (AQTA), an interactive web based visual analyses system support for the environment data. AQTA allows the seamless integration of predictive models and detailed patterns analyses visualisation. This interface provides back-and-forth dialogue with the designed multiple ML models and comparisons for better visual predictive assessments in different time conditions for chosen parameters. Moreover, AQTA provides data selection, display, visualisation of past, present, future and correlation structure among air parameters through various interactive charts, highlighting the predictive models' effectiveness. The findings from this technique corroborate the city's COVID lockdown (year 2020) conditions and sudden changes in patterns, highlighting the improvements in the pollutants concentrations. Further, this study also reveals that the decrease in the concentration of one pollutant does not ensure that the surrounding air quality would improve as other factors are interrelated. The AQTA can be further advanced by highlighting the locations of different sensors with an add-on to the sensor nature's monitoring and this motivates the following technique.

Geospatial Data Visual Assessments: (section 5.2) The last technique focuses on different environmental data geospatial locations (sensors locations). The unsupervised Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) clustering is applied on a series of (above mentioned) parameters to analyse the data trends. The HDBSCAN works well with noisy datasets. Furthermore, the ML transformer predictor is trained for modeling the future dominant (high measurements) locations with time as the output. The selected environmental data variations are compared and analysed in the spatio-temporal frame to provide detailed estimates on change in the average conditions in a region over the years.

Chapter 6 - Conclusion & Future Work: The last chapter of this thesis first summarises the contributions presented in this thesis. Then, the

conclusions of the approaches are discussed concerning the mentioned research challenges. The results obtained from these experiments show that the designed techniques are able to discover the temporal relationships among complex environmental data interactively in different time frames. Further, the thesis concludes with the summary of the techniques developed to inference the spatio-temporal environmental parameters, enabling the interactive exploration processes and recommendations for the future work.

2

Fundamentals & Background

The thesis presents visual assessments of time series prediction approaches designed especially for the environmental data. Furthermore, the presented approaches primarily focus on devising an accurate Machine Learning and deep learning framework that supports the user in understanding and comparing the model accuracy in relation to essential aspects of trends, time frame, and correlating together with considered meteorological and pollution parameters visually. This chapter provides the necessary foundations by introducing the general concepts of Visualsiation (section 2.1), an introduction to Visual Analytics (section 2.2), and GeoVisualisation and Visual Analytics applications concerning this thesis domain (section 2.3). Furthermore, in the section 2.4, the concepts of explainable AI, the introduction of Machine Learning and deep learning architectures concepts are discussed, followed by technical terminologies and concepts that are significantly used in this work (section 2.5, section 2.6, section 2.7, section 2.8). The accuracy measures which are considered in this work for evaluation are discussed in section 2.9.

Parts of this chapter have previously been published in:

Harbola, S. and Coors, V. (2018), 'Geo-Visualisation and Visual Analytics for Smart Cities: A Survey', *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-4/W11, https://doi.org/10.5194/isprs-archives-XLII-4-W11-11-2018, 11–18;

Harbola, S. and Coors, V. (2019a), 'One Dimensional Convolutional Neural Network Architectures for Wind Prediction', *Energy Conversion and Management*, 195, https://doi.org/10.1016/j.enconman.2019.05.007, 70–75;

Harbola, S., Storz, M., and Coors, V. (2021b), *Augment Reality for Windy-cities:3D Visualisation of future wind nature analysis in city planning* (Springer, (To appear, accepted on 2020- July -20));

Harbola, S., Koch, S., Ertl, T., and Coors, V. (2021a), 'Air Quality Temporal Analyser: Interactive temporal analyses with visual predictive assessments', *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, https://doi.org/10.2312/envirvis.20211083.

In the second part of this chapter, the concepts of predictions for time series data (with focus on environmental data) using Machine learning are discussed followed by the essential background work review (section 2.10). Based on comprehensive studies of the above mentioned topics, interactive prediction techniques to support users predicting the future of time series data visually is discussed in the following section 2.11. Moreover, this chapter also highlights how environmental data's Visual Analytics can blend in with advanced Machine Learning and deep learning (section 2.4) prediction models for creating applications for our surrounding that will increase environmental data sense making and awareness.

After the foundations have been laid, the chapter concludes with an introduction of visualisation and Machine Learning including deep learning advancements in time series predictions, outlining how Visual Analytics could comprehensively leverage information to enable task oriented environmental situation awareness visually.

2.1 Visualisation

The ease of analysing the data quickly and interactively is getting advanced daily, with increasing complexity for unstructured data analyses and representation that are challenging. There is a requirement for exploring the data insight to aggregate analysis for better technology intelligence (Card et al., 1999). Visualisation is an art of using computer supported, interactive visual demonstrations of the data in order to escalate user knowledge and comprehension gain. Extensive literature exists that explains how data needs to be operated to transform into user readable views, first extracting the relevant data aspects that shall be presented to the user, followed by mapping data with visual structure that the users can manipulate through interaction at any stage of view and operation (Shneiderman, 1996; Card et al., 1999). Thus, it aims at data transformation into a visual representation that is understandable for the users, helping them explore more of the data insights.

The following section first introduces GeoVisualisation and Visual Analytics on a conceptual level. Afterwards, the core components of Visualisation, elaborating the concept of data processing and analyses algorithms are elucidated, building the background of the techniques that are used in this thesis. Thus, it aids in explaining and simplifying the time series predictive analyses using Machine Learning concepts and their applications in combination.

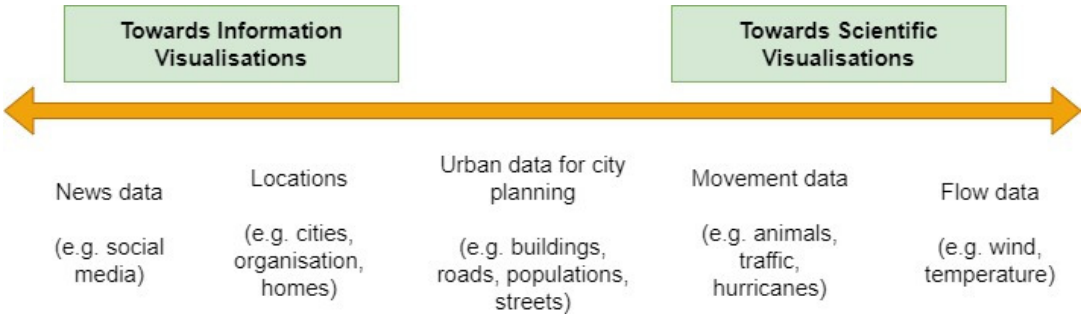


Figure 2.1: Real world examples of Visualisation MacEachren et al. (2013). (This figure is modified after MacEachren et al. (2013)).

2.1.1 GeoVisualisation

Data Visualisation can be categorised into three groups, namely, Information Visualisation, Scientific Visualisation and GeoVisualisation depending on the type of the data. Information Visualisation helps in Visualisation of abstract data, Scientific Visualisation covers spatial data and GeoVisualisation (GV) incorporates both the abstract and the spatial data. Figure 2.1 shows the real world examples that can be resolved using these techniques (MacEachren et al., 2013).

GV integrates approaches from Visualisation in Scientific Computing (ViSC), cartography, image analysis, Information Visualisation, Exploratory Data Analysis (EDA) and Geographic Information Systems (GIS) (Otto et al., 2010; Andrienko et al., 2011; Ramathan et al., 2013; Chen et al., 2014). A city's geospatial data is collected using traditional techniques like surveying, photogrammetry, sensors and techniques like Global Navigation Satellite System (GNSS), Light Detection and Ranging (LiDAR), Synthetic Apeture Radar (SAR) and Unmanned Aerial Vehicle (UAV), and can vary from small to large volumes (Komninos et al., 2013; Bhattacharya and Painho, 2017). Common types of the geospatial data are:

1. point data, $e.g.$, crime cases,

2. continuous values and discrete distributions, $e.g.$, earthquake readings, and

3. continuous values and continuous distributions, $e.g.$, climate simulation data.

Table 2.1 expands the various domains and corresponding examples of the georeferenced data.

Table 2.1: Georeferenced data present in various domains.

Domain	Example of georeferenced data
Daily life	Position, destination, routes
Demographics	Population, labor, crime rate by areas, regions
Urban planning	Growth rate, architecture, district types
Transportation/Logistics	Location of assets, delivery networks
Security/Intelligence	Location and movement of suspects
Medicine/Epidemics	Region of reported infections
Climatology/Meteorology	Weather, regional climate changes, pollution

2.1.2 Modules

GV consists of six main modules: data transformation and analysis, filtering, mapping, rendering and interactive user involvement. The collected large data either can be in structured form or as well as in complex form ($i.e.$, semi-structured, unstructured, spatial, temporal and multimedia). The data transformation and analysis are tasked with extracting the structured data from the large input data (MacEachren and Kraak, 2001; Thomson et al., 2005; Maciejewski et al., 2010; MacEachren and Kraak, 2011; MacEachren et al., 2012). For the complex form data, the data mining techniques like clustering can be used to extract the related structured data for visualisation (Southworth and Peterson, 2000). Filtering module corrects the structured data for noise by applying smoothening filters, for missing values by applying interpolation techniques and for measurement errors (Church and Cova, 2000; Johnson, 2004; Chudá, 2007).

These corrections automatically select the key data for visualisation (Arentze and Timmermans, 2000; Dasgupta and Kosara, 2011; Kitchin, 2011; Dasgupta et al., 2012; Bröring et al., 2014). After filtering, the data is mapped to geometric primitives like points, lines, regions and may have several attributes like colour, texture, position, and size. Users can transform the geometric data into image data using the rendering module and interact with the generated images through various interactive controls to explore and understand the data from different perspectives (Hofmann et al., 2012).

Moreover, interactive analysis and visualisation are driven by the applications and solutions in the domain it is applied (Trindade et al., 2017) and as a result, research in this field is usually motivated by real world user requirements and desired output (Claessen and van Wijk, 2011; Lloyd and Dykes, 2011; Chen et al., 2014; Cao and Cui, 2016).

Separate view	Space time combined view
Clear and causal interactions (navigate, select, manipulate)	Complicated 3D interaction in 2D environment
Gives an undistorted view on the geo-spatial. Clutter reduction, good visual scalability due to separation	Additional clutter and occlusion in 3D cube, bad visual scalability. Perspective distortion.
Spatio-temporal overview not available	Spatio-temporal overview available
Spatio-temporal correlation difficult to identify	Spatio-temporal correlation easily identified

Figure 2.2: The differences between the separate view and space time combined view, according to Harbola and Coors (2018).

.

2.1.3 Techniques

The techniques that can be used for visualising sensor's data are scatter plot, heat maps, height maps, survey plot, logic diagrams, parallel coordinates, multiple line graph, sammon plots and multi-dimensional scaling, polar charts, principal component and principal curve analysis, logic diagrams, choropleth maps, isolines, tilevis, plume chart, dashboards, quartile chart, trees, network and glyphs. Del Fatto et al. (2007) give schematized representations of territories $i.e.$, chorems, for visual summary of spatial databases.

Moreover, combining time and space provides temporal and geospatial correlation and helps in interactive temporal visualisation and examples of these are population development over time, epidemic spread over time and movements (traffic, animals, pedestrians, hurricanes, particles) (Sun et al., 2013; Sun and Li, 2016). Figure 2.2 gives the difference between visualising time, space separately (Figure 2.2, first column) and in combination (Figure 2.2, second column). Mapping time to space can be achieved using three methods,

1. separate views of each and interaction by brushing and linking,

2. space time cube where third axis represents time,

3. animation involving time.

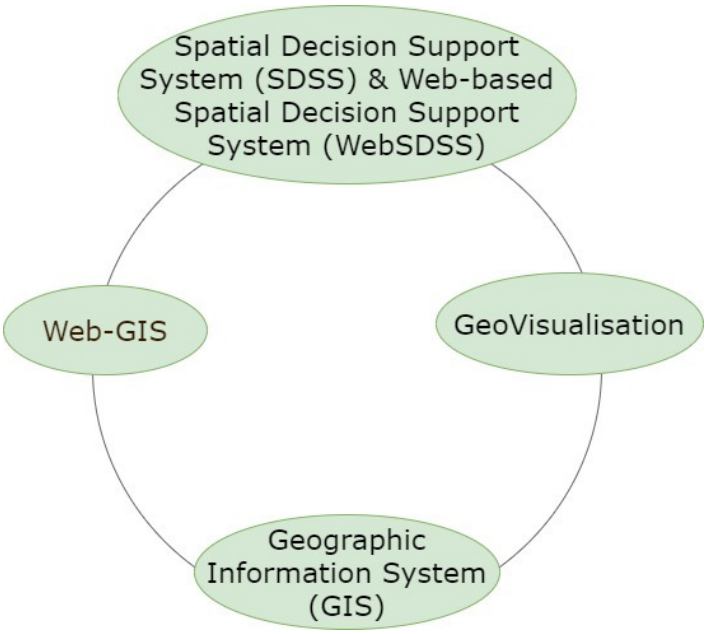


Figure 2.3: Data management and visualisation techniques and tools, according to Harbola and Coors (2018).

.

2.1.4 Data Management

The management of the geospatial data is also an important aspect for visualising the georeferenced data collected from the cities because different sensors measure more than one kind of data at a time and may cover a large area, thus displaying all sensors value is difficult. GIS manages geographically referenced information and aids in the geospatial data visualisation by analysing, managing and displaying the geospatial data and is also supported by Spatial Data Infrastructure (SDI) for both static and real time data (Figure 2.3). Integrating the GIS with the web ($i.e.$, Web-GIS) enhances the interactivity of users with maps and improves spatial analysis as shown in Figure 2.3 (Goodchild, 2007, 2013; Holliman et al., 2017).

The two types of geospatial data management models can be represented either by raster or vector data models (Huang and Liang, 2014; Stefan et al., 2017). The Open Geospatial Consortium (OGC) with its Senor Web Enablement (SWE) initiative passed the standards to control, detect and receive sensor data and some examples are, Sensor Observation Service (SOS) designed for 2D data and dynamic 3D SDIs for 3D data. OGC sensor web enabled open architecture makes it possible to handle most types of sensors (Prandi et al., 2013). Web based Spatial Decision Support System (SDSS/WebSDSS) helps to solve complex geospatial data problems relat-

ing to urban planning, site selection and decision making (Figure 2.3). A WedSDSS includes problem solver web based GIS and geographic data retrieval facilities, analysis and display (Sugumaran and Sugumaran, 2013). Recently multidimensional distributed spatial platform integrating sensor web with SDIs, Smart Cities Intelligence System (SMACiSYS) has been developed (Bhattacharya and Painho, 2017).

2.2 Visual Analytics

The Visual Analytics (VA) became popular and more advanced with Thomas and Cook (2005) work, where the research and development agenda for Visual Analytics was explored. They have elaborated VA as "the science of analytical reasoning facilitated by interactive visual interfaces" (Thomas and Cook, 2005). Moreover, VA combines visualisation with data processing and algorithms assessments using domain expert's experiences and knowledge with machine advancements and the system's potential to process extensive datasets into meaningful information quickly (Thomas and Kielman, 2009). The VA concept is further explained in the following section along with visualisation core components.

2.2.1 General Concept

Visual Analytics aims to efficiently use a large volume of information in various applications by adequately merging the strengths of intelligent automatic data analysis with the user's visual perception and analysis capabilities interactively. Figure 2.4 provides a glimpse of the Visual Analytics workflow cycle as described by Keim et al. (2010). Visual Analytics provides two broad ways for applications and problem solving. Firstly, it could be intended to give the user the capability to extract insights from the data to help interactive user decision making (Pirolli and Card, 2005). Creating a workflow to gain insights from the available data presents an apparent attempt for such a sense making process. Secondly, it could help to present concept workflow in order to provide the ability to enable data processing that combined user readable visualisations along with automatic data processing (Thomas and Cook, 2005).

Sacha et al. (2014) provided the concept of combining these two ways into a knowledge generation model that incorporated the advantages of both ways. The available data is processed, followed by visualisation and data models are generated; the experts can examine that to analyse the data. Different stages of knowledge about the data are acquired as an outcome of the performed analyses (Sacha et al., 2014). During the analyses,

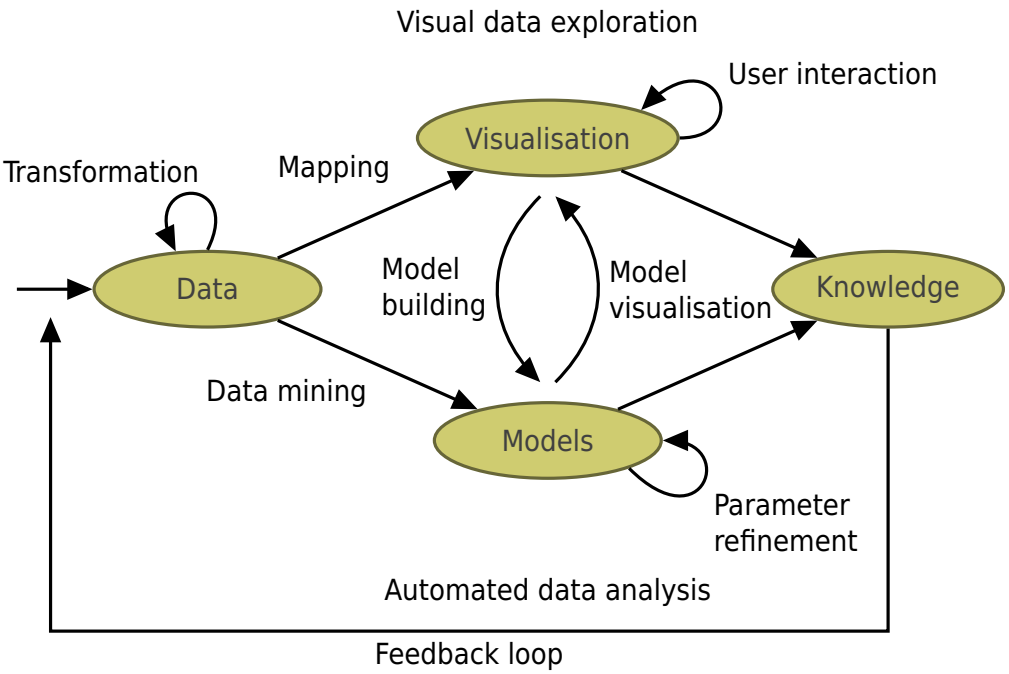


Figure 2.4: The visual analytics cycle according to Keim et al. (2010). (This figure is modified after Keim et al. (2010)).

the user could extract findings from the data. If findings are already known or are of no relevance, the study continues. While if the findings were relevant, they become insights that need to be verified. This verification could be performed by formulating a hypothesis about the understanding and continuing the analysis with regard to the hypothesis and upgrading. Furthermore, if the data supports the hypothesis, the insights becomes knowledge that may also be transferable by the user analyst to other data sets or situations.

2.2.2 Visual Analytics Importance

Digital visualisation gives an effective medium to analyse, but it is Visual Analytics (VA) that aids in the design of the cities (Marsal-Llacuna, 2015). VA transcends the pictorial representations and links the various tasks to appeal visually as well as reflects the quality and efficacies of the urban design (Gröger and Plümer, 2012; Albino et al., 2015). Thomas and Cook (2005) have explained a coordinated technical vision for government and research investments and help to ensure that a continual stream of technology and tools enters analysts' hands and answers related crisis queries accordingly. Moreover, providing many advancements of services available

in the cities and the corresponding use of VA are discussed below. By categorising the services into seven broad categories, VA helps explore and gain insight into each section. VA enhances building services to maintain and manage cities' assets, providing asset performance index and other optimal intervention point analytics. This makes the information transfer more transparent by connecting and involving citizens with interactive visual interfaces while gaining citizens' satisfaction levels and citizens awareness levels index. Visual assessments of data quality index, transportation conditions index, traffic forecast help in supporting cities infrastructure based on sensors services. Cities services like smart land use analyses are improved with observed rates for different land uses and travel between zones, land value transportation index, and zone accessibility index visually. Furthermore, VA supporting business models strategies and partnering services, helps to resolve queries by answering the percentage of private sector investment, number of partnerships, improvement in service delivery, private public sector interaction and money invested (Prieto, 2013). Morover, VA enhances the urban automation by supporting a lot of work investigating the percentage of automated vehicles within the entire citywide convoy, the percentage of automated vehicles in use by city public and private groups, the proportion of deliveries made by automated vehicles, and the proportion of passengers carried by automated transit. The services related to user centric mobility provide knowledge on the citywide mobility index, user satisfaction index, and reliability index of transportation service delivery.

In designing an interactive urban VA, there are generally four essential features of VA, involved in the smart design of a city. First is the GeoVisualisation of the city design in 3D or 2D maps and transforming into several virtual environments to aid city designers and users to experience the design (Fu and Zhang, 2017). The second feature is the layout of the networks for understanding the interaction among users and their movement (Tan et al., 2017). The third feature involves social media that reflects the users' communication in the real and virtual design of the city (Ahvenniemi et al., 2017). The fourth feature deals with the planning process based on the online information where users contribute to the improvement of the designs and generating more data (Kumar and Prakash, 2016).

2.3 GeoVisualisation & Visual Analytics Applications

GV and VA tools and methods encourage collaboration and communication between entities and provide services to many sectors in the smart city, as well as improve customer's experiences and business opportunities

(Hashem et al., 2016). The solutions found in the recent literature can be classified into the following categories: smart grids, smart healthcare, smart transportation, smart governance.

Smart grids have enabled researchers to integrate, analyse, and use real time power generation and consumption data, as well as other types of environmental data (Nga et al., 2012; Tsolakis and Anthopoulos, 2015; Sanchez and Rivera, 2017; Stefan et al., 2017).

Smart healthcare related analytics tools allow healthcare specialists to collect and analyse patients' data, which can likewise be used by insurance agencies and administration organisations. Moreover, proper analytics of large healthcare data can help predict epidemics, cures, and diseases, as well as improve quality of life and avoid preventable death (Noon and Hankins, 2001; Tunio et al., 2017).

Smart transportation provides VA applications to visualise and analyse a large amount of data collected from transportation system, thereby helping in the improvement of the transportation systems in terms of minimising traffic congestion, by providing alternative routes and reducing the number of accidents through the analyses of the history of mishaps, including factors such as their cause and the driver speed (Andrienko and Andrienko, 2013; Kalamaras et al., 2018). Singh et al. (2016) developed a framework of interactive VA for detecting bike riders without helmet automatically in city traffic.

Smart governance data analytics can help governments establish and implement satisfactory policies taking into consideration the needs of the people in terms of health, social care and education. In addition, the ratio of unemployment can also be reduced by analysing the large data of different educational institutes (Lara et al., 2016; Wang et al., 2017b). Kohlhammer et al. (2010) developed information visualisation and VA for governance and policy modelling. Similarly, the Trento i-scope project deals with citizen participation for web based services, giving an interoperable framework for the visualisation and processing of 3D city models on mobile devices. Smartmap Berlin provides visualisation and analysis of Berlin in photorealistic 3D format. Strengthening the smart governance application attempts of the city modelling in focus to match with the emerging practices of eco-town based urban developments have been implemented in Germany, Netherlands, Sweden and example smart city initiatives from Korea (Bayulken and Huisingh, 2015; Yigitcanlar and Kamruzzaman, 2018).

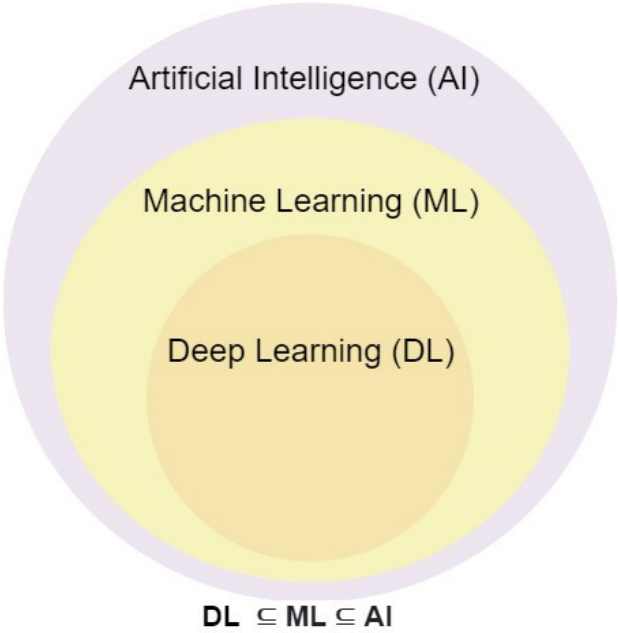


Figure 2.5: Relationship between AI, ML, and DL, according to Goodfellow et al. (2016). (This figure is modified after Goodfellow et al. (2016)).

2.4 Machine Learning Fundamentals

The studies involved in exploring the various ways to build an intelligent mechanism or program that would help solve problems creatively, which usually requires human intelligence, comes under Artificial Intelligence (AI). Machine Learning (ML) is the subset of AI, a general term to define when computers learn from data. Algorithms are designed to recognise patterns in the data and help make predictions for new data. Furthermore, Deep Learning (DL) is a subset of Machine Learning where algorithms are based on a hierarchy of neural networks. A large number of features are automatically learnt by these deep learning algorithms during the training phase instead of manually designing them, in contrast to basic ML algorithms $e.g.$, linear regression, decision trees, and Support Vector Machine (SVM) $etc.$ where features are manually designed additionally if required. Figure 2.5 shows the relationship between AI, ML, and DL, highlighting that ML is a type of AI, while DL is an important complex part of ML (Goodfellow et al., 2016).

Explainable Artificial Intelligence (XAI) represents methods that can be of different formats, such as rules, numerical, textual or visual information for interpreting ML algorithms (Miller, 2017; Vilone and Longo, 2020; Xie et al., 2020). These methods can comprise basic toolkits, complex techniques, and interactive and simple visual interfaces. XAI can be broadly

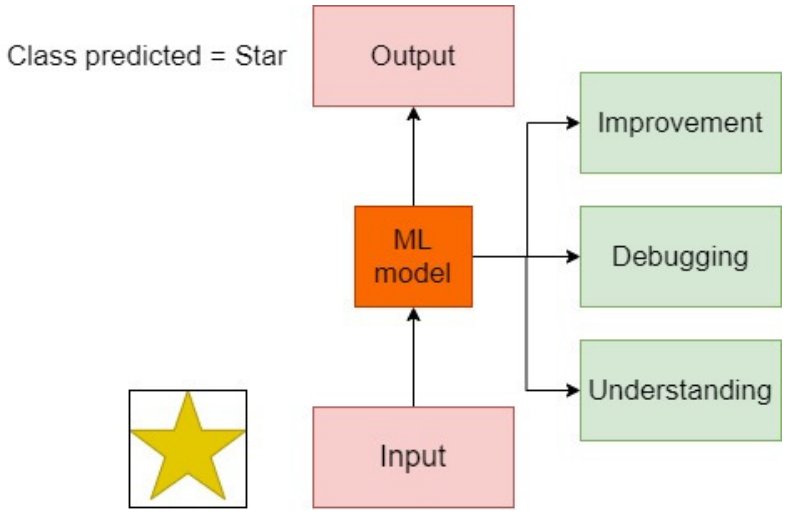


Figure 2.6: XAI overview.

based on the understanding, debugging, and refinement at the ML algorithm design stage or after the algorithm has been trained and tested in order to perform results interpretability and comparison (Choo and Liu, 2018). Here, the focus of explanation can involve anyone of this motive ($i.e.$, either for understanding, debugging and refinement) or all these purposes together to enable the user to manage, understand, and appropriately trust the developed ML model, thus contributing to the decision making process towards XAI. This decision making process can be based on the methods focused on producing visual representations of the ML models, their delivered output, intermediate results, tunning parameters, accuracy metrics and architecture insights using basic graphs such as heat maps and bubble charts.

The Figure 2.6 explains the concept of XAI, where an image's class prediction is performed. Image input is given to the designed ML network that classifies the input image's class (here the class is the star). Now to debug, refine and understand this developed model during the training time as well as after the model has been trained for performing the testing and inference process would require some visual representations and thus contributing towards the domain of XAI. For example, a user constructs simple line graphs and histograms of desired parameters, such as the accuracy, loss value of a particular neuron, that offer a simple representation of the low level (describing in detail the individual components) information of the ML model. Further, XAI can involve a user in the analysis process using his domain knowledge and expertise to increase understanding of the ML techniques and intuitively analyse that the model follows the rules, thereby building trust. This analysis process can use VA to interact with

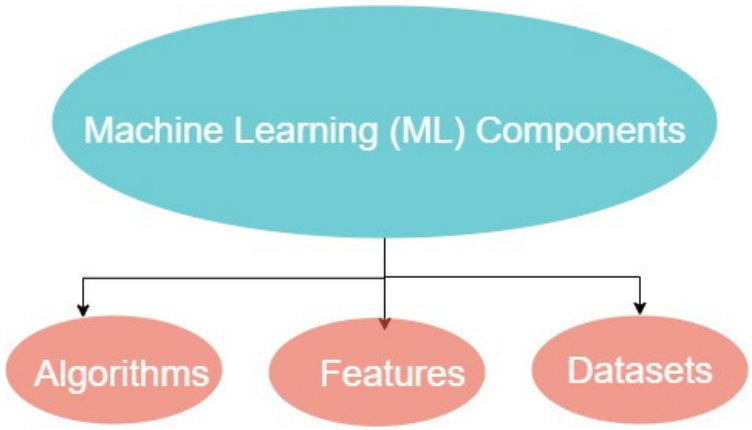


Figure 2.7: ML components overview.

ML models at various stages (Choo and Liu, 2018).

Furthermore, the important motive behind using ML is to create algorithms that learn and make predictions on the desired datasets. In order to make the machine learn and perform, essential starting components based on datasets, features and algorithms, are required. Figure 2.7 shows the ML components graphically.

Datasets: The collections of samples on which the ML systems are trained are known as datasets. The samples can include any kind of data (images, text, numbers); moreover, historical temporal environmental data are considered for this thesis work samples creations.

Features: The essential properties or example of data used as the key for solving the problem, explaining to the machine where and what are the crucial aspects to pay attention to are called features.

Algorithms: The algorithms are designed in order to solve the task. Multiple algorithms can be used to solve one task. Moreover, depending upon the accuracy achieved, resources utilised, desired results, and total execution time taken, algorithms assessments can be performed and appropriate algorithm can be selected.

Furthermore, a neural network consists of some layers, where the first layer is known as the input layer, the last layer is defined as the output layer. In contrast, the intermediate layers are known as hidden layers as their values are not determined in the training set and computed by the network itself. Each layer consists of an array of neurons, which have their own weights and biases. If each neuron of one layer is connected to every

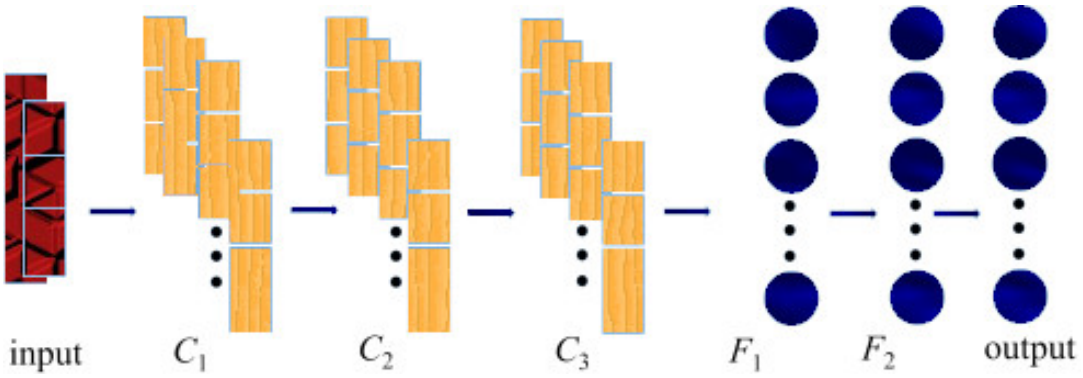


Figure 2.8: The 1D Convolutional Neural network (CNN) architecture (1DS). (This architecture is designed in this work and explained in detailed in the following chapter 3).

other neuron of the previous layer, that layer is called a fully connected layer (Nielsen, 2015).

To explain these terms in more details with respect to this thesis domain, suppose the designed deep learning architecture is similar to as shown in Figure 2.8, that represents an One Dimensional (1D) Convolutional Neural Network (CNN). A CNN consist of some convolutional layer, fully connected layer, input and output layer. In this figure the leftmost layer (parallel plates) represents the input layer while rightmost (circles array) represents the output layer. The intermediate layers are three convolutional layer (C_1, C_2, and C_3) and fully connected layer (F_1 and F_1). To make an easy graphical understanding of the convolutional layer and the fully connected layer are represented with vertical rectangular plates array and vertical circles array, respectively, in Figure 2.8.

Moreover, a convolutional layer performs a convolution operation, a linear function that involves the multiplication between an array of input data and a 1D array of weights, called a kernel. A dot product (element-wise multiplication) is applied between the kernel size input and kernel, resulting in a single value (because of the summation). The kernel size is kept smaller than the input as it helps in multiplying several times the input array at different places with the same kernel (set of weights) in the input. The number of steps the kernel moves in each convolution step is controlled by stride. For 1D CNN, this is equal to 1. More details about the CNN layers insights are discussed later in the following sections. The obtained output array after applying the convolutional task is called a feature map (Andrade, 2019). This feature map (each value) become input for the following successive layers. All layers of CNN comprise of multiple neurons; for the input layer, this number is equal to the input matrix for

$e.g.$, in a Neural Network if the number of neurons in the input layer is seven, then it's each sample expected to have seven values. Moreover, for all the layers (excluding the input layer), the activation function determines how each layer's neurons pass the sum of weighted input, act on it, and transform this input (weight and bias) into output. There are various types of activation functions, and below are discussed some of the activation functions used in this thesis work.

Activation Functions

Sigmoid: A sigmoid activation function is a logistic activation function that takes any possible real value type input and transforms this into 0 to 1 range output values. The more positive is the input, the closer the output value is to 1, whereas the more negative the input, the closer the output is to 0.

Tanh: A tanh activation function is a hyperbolic tangent activation function that takes any real value as input and outputs values in the range -1 to 1. The more positive the input, the closer the output value is to 1, whereas the more negative the input, the closer the output is to -1.

ReLU: A Rectified Linear Activation Function (ReLU) activation function transforms the input if negative into 0; otherwise, the unmodified input value is returned. This function also overcomes the limitations of Sigmoid and tanh.

ELU: An Exponential Linear Unit (ELU) activation function overcame some of the problems of ReLUs and inherited some of its positive qualities. The function transforms the negative input into a value slightly less than 0; otherwise, the original input value is returned. To explain it further, suppose, if inp defines the input value, then the mathematical representation of the ELU function is given as: if inp > 0 then ELU (inp) = inp, else if $inp \leq 0$ then ELU (inp) = ELU (inp) + α For this activation function, an α value is picked commonly between 0.1 and 0.3.

2.5 Layers of Convolutional Neural Network

The Convolutional Neural Network (CNN) architecture has an input layer followed by multiple consecutive convolutional layers. Successive convolutional layer followed by multiple fully connected layers. Moreover, the last fully connected layers act as an output layer $i.e.$, the softmax layer

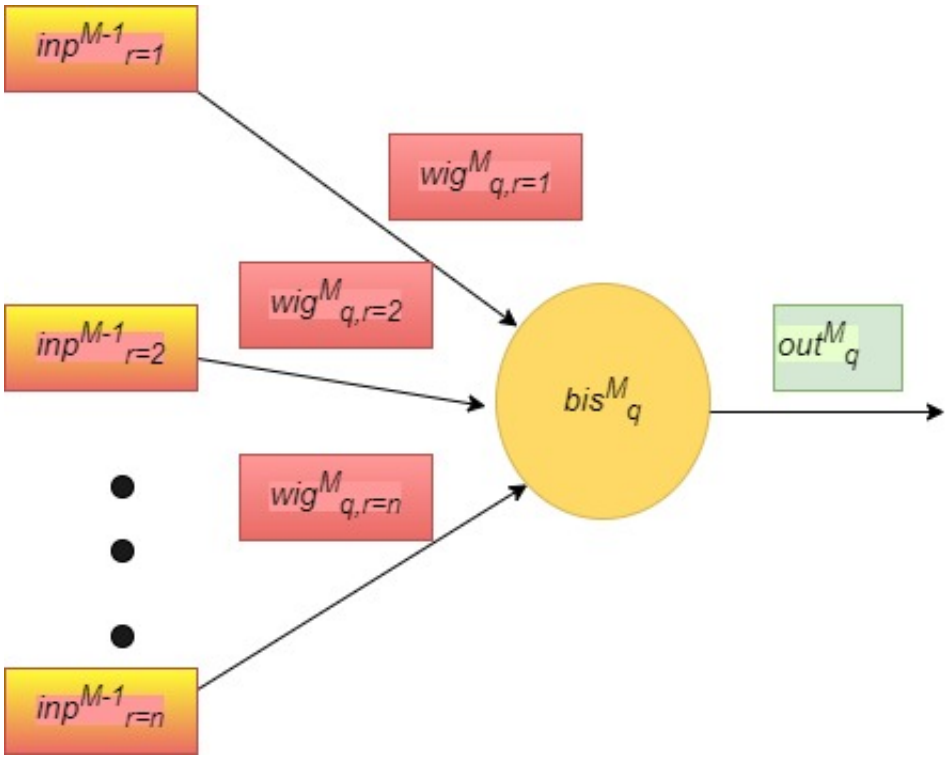


Figure 2.9: The structure of a single neuron.

(Krizhevsky et al., 2012). Multiple feature maps in the convolutional layers enable finding various patterns at various locations of the datasets. The neurons of subsequent fully connected layers and convolutional layers are connected differently. Each feature map in a convolutional layer is spatially related to all the feature maps in the previous layer. A convolutional layer feature map consists of point (1D) or a pixel (in 2D) which takes input from all the points lying on the previous layer feature maps, for the same kernel portion at the same spatial position. The training of the CNN architecture is performed using the backpropagation algorithm. In order to comprehend the weight and biases of the neurons, during network training, several samples whose inputs and corresponding outputs are known are used. The learnt weights and biases of the trained network are used to estimate the classes of unseen testing and inference samples which are different from the training samples. The concept of the layers, softmax, functions, feedforward and backpropagation steps are elaborated in the following section.

In a typical convolutional neural network, the feed-forward and back-propagation steps play a crucial role. Calculating the value of the neuron from input layer to output layers are performed as the feed-forward step. While computing the error of neurons with simultaneously updating their

weights and biases from the output layer to the input layer is defined as a back-propagation task (Andrade, 2019). Furthermore, expanding the terminologies mentioned above the neuron (neu) in a specific layer can be mathematically represented as follows (as shown in Figure 2.9), if ${neu_q}^M$ represents q neuron in the layer M with multiple inputs ${inp_r}^{(M-1)}$ (where, $r \Rightarrow 1...n$, and output of r neuron after applying an activation function avt in layer $M-1$), ${bis_q}^M$ a single bias and ${out_q}^M$ corresponding output (Nielsen, 2015). ${wig_{q,r}}^M$ defines the associated weights with the inputs ${inp_r}^{(M-1)}$ ($r \Rightarrow 1...n$). The weighted sum of the inputs is given by output ${out_q}^M$ and can be written in the form of an equation, as shown in equation. 2.1, where "." denotes the matrix multiplication. An activation function avt, for example RELU, sigmoid, and ELU (as discussed in section 2.4), any can be applied on ${out_q}^M$ in order to obtain ${inp_r}^M$ ($i.e.$, ${inp_r}^M$ = avt(${out_q}^M$).

$$ {out_{\mathrm{q}}}^M = \sum_{r=1}^{n} {wig_{\mathrm{q,\,r}}}^M . {inp_{\mathrm{r}}}^{M-1} + {bis_{\mathrm{q}}}^M \tag{2.1} $$

Unlike convolutional layers, the fully connected layers have multiple linearly arranged neurons. Each neuron of the preceding fully connected layers is connected to every other neuron in the next fully connected layer. Considering that the fully connected layer $M-1$ each neuron is connected to every neuron in fully connected layer M, and there are n_{M-1}, n_M neurons in each layer, respectively. Equation. 2.2 denotes the feed-forward step between fully connected layers, to obtain the output ${out_{FC}}^M$ (a 1D matrix consist of ${out_q}^M$ ($q \Rightarrow 1...n_M$) (Nielsen, 2015).

$$ {out_{FC}}^M = {wig_{FC}}^M \cdot {inp_{FC}}^{M-1} + {bis_{FC}}^M \tag{2.2} $$

Moreover, the motive behind the back-propagation step is to update after each iteration of training the weights and biases. This process calculates the output layer neurons' error first and then compute intermediate layer neurons' error (back-propagation, $i.e.$, the error in the output layer is propagated step by step towards the input layer). Estimating the weight and biases changes using the calculated error and updating the weight and biases continuously. Information about all the associated neurons in the preceding and successive layers and their corresponding weights and biases are utilised during these calculations.

Softmax

In the neural network classification approach, most of the time, the activation function avt used in the output layer is the softmax function. Thus the output layer using the softmax function is also called the softmax layer.

However, for the non-output layer, the activation functions can be any activation function (sigmoid, tanh, ReLU, and ELU) other than softmax. The softmax layer provides different classes probabilities. Typically in classification, the output layer neurons' number is equal to the number of classes in which the input is required to be classified. Therefore, corresponding to n_m classes, there must be n_m neurons in the last fully connected layer. Before applying the softmax function, the output of each neuron in the last fully connected layer is denoted by ${out_q}^M$ ($q \Rightarrow 1...n_m$) as given in equation. 2.2.

$${inp_q}^M = \frac{e^{{outq}^M}}{\sum_{r=1}^{n_m} e^{{outr}^M}} \tag{2.3}$$

The softmax function is applied to this output as defined by equation. 2.3, where exponential value of ${out_q}^M$ is represented as $e^{{outq}^M}$. The sum over all the neurons is denoted by $\sum_{r=1}^{n_m} e^{{outr}^M}$ and corresponding softmax output of a neuron is given by ${inp_q}^M$ (Nielsen, 2015). Furthermore, $\sum_{q=1}^{n_m} e^{{outq}^M} = 1$ (as it can be computed from equation. 2.3), thus ${out_q}^M$ delivers the probability of a class.

2.6 Cross Validation

One of the essential tasks is to identify the best way to split the datasets into training and testing samples. The evaluation of the predictive model, where one subset (samples) of the datasets is used to train the model, and the remaining samples are used to test the model, is performed by using cross validation. A k-fold cross validation technique is also used in this work. Suppose there are A to I (A, B, C, D, E, F, G, H, I) samples in the dataset, and k = 3 ($i.e.$, 3-fold cross validation) is applied. Now for a simple explanation of the k-fold cross validation concept, the training and testing samples would look like as shown in Figure 2.10. After iterating this step multiple times (here iteration = 3), the results are summarised and aggregated, comparing all the iterations. Therefore in the above example, three repeats of 3-fold cross validation are performed, meaning that 3-fold cross validation is applied three times, fitting and evaluating models on the dataset.

2.7 Regularisation

The situation in which the model cannot predict well for the training data and performs poorly with respect to the accuracy metrics raises an underfit model and requires to be fixed. Furthermore, the condition in which

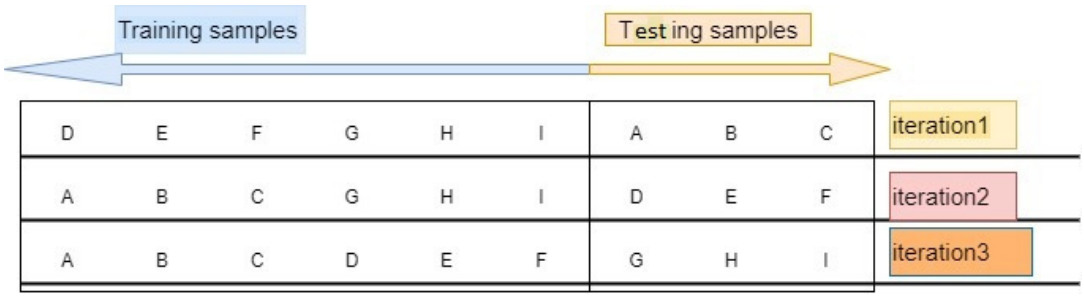


Figure 2.10: Splitting the dataset using 3-fold cross validation.

the model works well during the training time while in the testing time loses its ability and cannot predict test data. This raised problem and is defined as over-fitting. In order to avoid these situations, regularisation techniques are used, where the weights and biases of the neurons are optimised to obtain a well fitted model. L2 regularisation, dropout, batch normalisation, data augmentation (viz, Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling Approach (ADASYN)) techniques are used in this work to avoid over-fitting and under-sampling.

2.8 Data Preparation & Feature Engineering

The success of Machine Learning algorithms depends on how the data is presented and designed for these models. Transforming the available unstructured data into meaningful Machine Learning inputs is the requirement of each prediction analyses (Liu and Motoda, 1998). Learning a solution to a problem from the input data making Machine Learning algorithms perform successfully requires feature engineering. A feature could be defined as the representation of the information of available data into a format that would best fit the designed model's requirements and will solve the problem for which it is designed (Guyon et al., 2006).

2.8.1 Feature Importance & Selection

It is essential to estimate the usefulness of features objectively. The features designing are carried out carefully, evaluating the feature's effectiveness with the algorithms used.

Correlation Coefficient

The degree to which the two variables are related to each other can be discovered with the help of correlation analyses. Calculating the correla-

tion coefficients helps identify the extent to which the two variables are dependent on each other and their relationships. To evaluates the linear relationship between the two continuous variables, Pearson Correlation Coefficient (PCC) is used. Spearman correlation is often used to assess relationships involving the dependency between two continuous or ordinal variables (Ward, 1963).

Euclidean Distance

The most used and simple technique of measuring similarity is to calculate using the Euclidean distance between the points. For data with n dimensions, the distance of two points A_j and B_j is calculated as shown in equation. 2.4

$$D(A_j, B_j) = \sqrt{\sum_{j=1}^{n} (A_j - B_j)^2} \tag{2.4}$$

where A_j and B_j indicate the value of the j-th dimension points.

2.8.2 Feature Extraction

Reducing the dimensionality of the highly voluminous unstructured data to be modelled by predictive algorithms directly requires feature extraction techniques (Liu and Motoda, 1998).

Unsupervised Clustering

Unsupervised clustering identifies similar data points as distinct groups ($i.e.$, clusters) in unstructured datasets by finding the similarities among the points and grouping them with their shared similar properties (Goy et al., 2021). Some of the techniques used in this thesis to analyse the temporal time series dataset are discussed below. The two types of clustering techniques categories can be defined as flat and hierarchical clustering. Firstly the flat clustering partitions the data into clusters based on each cluster individually (where each cluster centre $i.e.$, centroid) and assigns the points to the specific cluster ignoring the interrelationships between the clusters. K-mean clustering is an example of centroid based clustering. Secondly, hierarchical clustering provides clustering at varying levels of details with introduced tree-like structures, describing the relationships between the clusters. Moreover, the hierarchical clustering can be sub categories into two types

1. Divisive (top-down) clustering $i.e.$, initially, all the data points are considered as one giant cluster known as the root cluster. Then recursively, the root gets divided into a set of child clusters, and then each child cluster further splits until each cluster with only a single point remains.

2. Agglomerative (bottom-up) clustering uses the concept of "dendrogram", constructed from the bottom level by combining the similar pair of clusters till all the data points are integrated into a root cluster $i.e.$, a single cluster at the end, out of clusters similarities.

2.9 Accuracy Measures

Confusion Matrix

The performance of the machine learning algorithms can be measured by a table called the confusion matrix (C). A table (matrix) where each row and column of the matrix represents the instances of an actual class and predicted class, respectively (Congalton and Green, 2008). Figure 2.11 shows the confusion matrix's graphical structure. The accuracy measures used in this work to evaluate the model efficacy are total accuracy, precision and recall. In order to explore the concept with an example, suppose the dataset contains samples from five classes (dog, tree, cat, rabbit, ground), and a confusion matrix is constructed to compute the accuracy metrics of the designed algorithm as shown in Figure 2.11. Here the column total, row total, number of samples correctly identified for a class x and number of total samples of all the classes are represented by C_{xy}, C_{yx}, C_{xx} and T_{sample}, respectively. The total accuracy is defined as the number of correct predictions obtained out of the total number of predictions performed by the algorithm. The number of correct cases where the algorithm correctly predicted the appropriate class x out of the total actual results is called precision (as shown in equation. 2.5).

$$Precision_x = \frac{C_{xx}}{\sum_y C_{yx}} \tag{2.5}$$

The number of cases where the algorithm correctly predicted x out of the total cases which are predicted is called recall (as shown in equation. 2.6).

$$Recall_x = \frac{C_{xx}}{\sum_y C_{xy}} \tag{2.6}$$

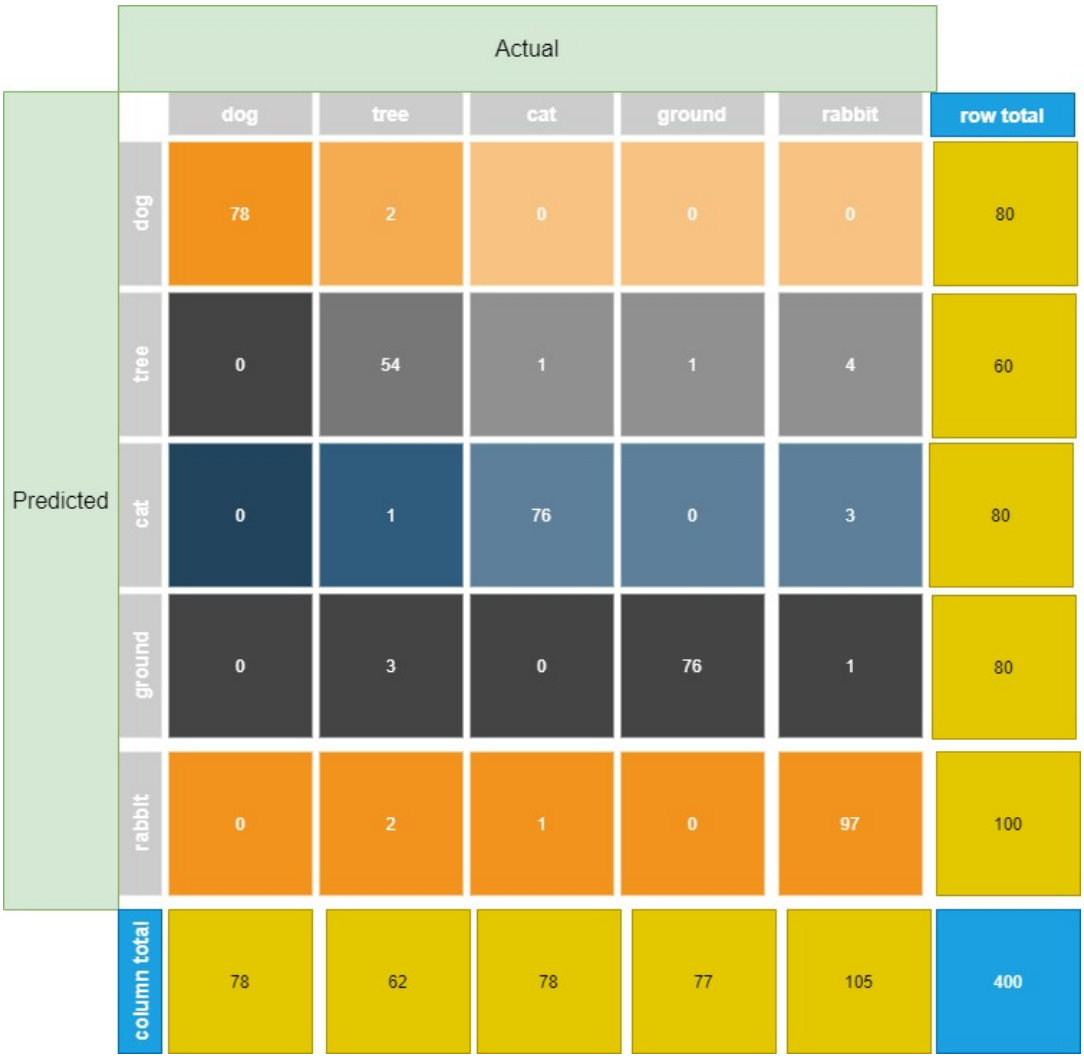


Figure 2.11: Table of a confusion matrix.

Using the above equations the precision and recall of dog class are given by equation. 2.7 and 2.8.

$$Precision_x = \frac{78}{80} \tag{2.7}$$

$$Recall_x = \frac{78}{78} \tag{2.8}$$

Precision and recall of tree class are given by equation. 2.9 and 2.10.

$$Precision_x = \frac{54}{60} \tag{2.9}$$

$$Recall_x = \frac{54}{62} \tag{2.10}$$

Similar calculations are repeated for rest of the other classes outputs precision and recall assessments.

Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error is utilised to compare predictions on different scales while expressing in percentage and is given by equation. 2.11, where M, $actual_t$ and $predict_t$ are equal to mean absolute percentage error, actual and predicted value, respectively. The t represents the data time index, and n represents the total number of data points fitted or predicted.

$$M = \frac{1}{n}\sum_{t=1}^{n} |\frac{actual_{\mathrm{t}} - predict_{\mathrm{t}}}{actual_{\mathrm{t}}}| \tag{2.11}$$

Symmetric Mean Absolute Percentage Error (SMAPE)

The Symmetric Mean Absolute Percentage Error overcomes the weakness of MAPE by expressing in percentage with upper and lower bounds of 0% and 200% respectively and is given by equation. 2.12, where M, $actual_t$ and $predict_t$ are equal to symmetric mean absolute percentage error, actual and predicted value, respectively. t represents the data time index, and n represents the total number of data points fitted or predicted.

$$M = \frac{100\%}{n}\sum_{t=1}^{n} \frac{|predict_{\mathrm{t}} - actual_{\mathrm{t}}|}{|actual_{\mathrm{t}}| + |predict_{\mathrm{t}}|} \tag{2.12}$$

2.10 Environmental Data & Models

An important source of renewable energy is the wind energy which is widely used as a green source of electricity generation (Vargas et al., 2010; Colak et al., 2012). The speculative nature of the wind makes its modelling more vital and challenging (Tarade and Katti, 2011; Lawan et al., 2014; Marović et al., 2017). The past states of the wind speed, direction, temperature, altitude, pressure, and other factors affect the behavior of the future wind trend. Further, the installation of the new wind turbines and sensors over a location requires prior assessment and prediction of the nature of the wind and is highly dependent on the impact of the wind speed and direction (Reed et al., 2011; Aissou et al., 2015). The time scales used for the wind speed, and direction prediction can be grouped into four categories, namely, very short term (these predictions include for a few seconds to 30 minute ahead), short term (include predictions from 30 minute to 6 hours), medium

term (predictions for 6 hours to 1 day ahead) and long term (from 1 day to 1 week predictions) (Yesilbudak et al., 2013; Yesilbodak et al., 2017). Very short term and short term wind speed and direction prediction reduce fluctuations and sudden cut-off in voltage and frequency due to variation in wind power and excessive wind speed (Miranda and Dunn, 2006). The medium term prediction of the wind parameters helps in maintaining an anticipatory control of the wind sensors (Kusiak et al., 2009b; Daraeepour and Echeverri, 2014) and for the online monitoring alerts (Zhou et al., 2011; Filik and Filik, 2017). Meanwhile, the long term wind speed prediction is of interest for the management of the energy distribution (Louka et al., 2008).

The wind prediction models can be based on firstly Numerical Weather Prediction (NWP) models (Louka et al., 2008), secondly Machine Learning (ML) (Sapronova et al., 2016), and thirdly the combination of both NWP and ML (Vladislavleva et al., 2013). NWP approach is based on the physical kinematic equations that use multiple meteorological variables which are necessary as input for the prediction model and operates by solving the complex mathematical models (Zhou et al., 2011; Filik and Filik, 2017).

In ML various concepts can be used such as fuzzy logic (Monfared et al., 2009; Martínez-Arellano et al., 2014), Artificial Neural Networks (ANN) with several hidden layers (El-Fouly and El-Saadany, 2008; Yesilbodak et al., 2017), and statistical models (Miranda and Dunn, 2006; Louka et al., 2008). Daraeepour and Echeverri (2014) presented a multi variable model for wind speed and power prediction in an hour for the next day. The model first filtered out the data by selecting the best set of inputs features and then used ANN to predict the successive values of the wind. Regression models using neural networks along with techniques like particle swarm optimization, wavelet transform (Martínez-Arellano et al., 2014; Wang et al., 2017a; Liu et al., 2018), REP tree, M5P tree, bagging tree, K-Nearest Neighbor (K-NN) algorithm (Jursa and Rohrig, 2008; Kusiak et al., 2009a; Kusiak and Zhang, 2010), principal component analysis, moving average models, Markov chain (Kusiak et al., 2009b; Vargas et al., 2010; Treiber et al., 2016), have been used for wind analysis. An n-tupled inputs was used to predcit the wind speed using K-NN classification by Yesilbudak et al. (2013). They have analysed the effects of distance metrics, nearest neighbours and input parameters. Support Vector Machines (SVM) and its variation (Kang et al., 2017), Least Square Support Vector Machines (SVM) (LSSVM) have also been used for forecasting wind speed (De Giorgi et al., 2014, 2016). Yuan et al. (2015) developed a short term wind power prediction hybrid model, based on LSSVM and gravitational search algorithm. The gravitational search algorithm was used to optimize the parameters of LSSVM. Different kernel function of LSSVM and their effects on wind power prediction was presented.

Recently, in the computer vision domain, deep learning algorithm using Convolutional Neural Networks (CNNs) have shown promising results for Two Dimensional (2D) images and Three Dimensional (3D) CAD models classification (Krizhevsky et al., 2012; Long et al., 2015; Szegedy et al., 2015). CNNs with multiple convolutional layers and fully connected layers, perform better than the traditional methods and learn their features automatically instead of manually designing them (Kuo, 2016; Qi et al., 2016). ANN with multiple hidden layers lack convolutional layers and are thus, unable to extract features unlike CNN (Long et al., 2015). Moreover, multiple CNN with several input views improve upon the single CNN accuracy with one view (Jung et al., 2019). Further, single CNNs (1D, 2D) have been used for temporal wind dataset to predict wind power and wind speed (Wang et al., 2017a; Liu et al., 2018), but by converting the One Dimensional (1D) temporal wind data into wavelets decomposition and 2D image information, thereby, losing the original 1D wind information. Liu et al. (2018) used wavelet packet decomposition for dividing the data into high frequency and low frequency data. The high frequency data was predicted using the CNN with 1D convolution operator, while low frequency data was predicted using CNN and Long Short Term Memory (LSTM) combination as a regression model, and the input dataset was smoothed. Moreover, these models for wind prediction have used limited datasets to analyse their models and predict only a few values ($e.g.$, 3) in the future. However, a single and multiple 1D CNNs that would work directly on the original 1D temporal wind dataset, without any transformation, with large historical wind dataset and with user defined time frame of prediction in future, are still required.

2.11 Time Series Visual Prediction

Temporal datasets are essential and measured across almost all the domains including environmental, healthcare, scientific and financial. Visual analytics (VA) supported with Scientific or Information Visualisation (Sci-Vis or Info-Vis) techniques are in demand and also crucial for analysing these time series datasets patterns (Aigner et al., 2011). The characteristics of data, its size, multi-dimensionality, and distribution contribute to make situation assessment one of the most demanding tasks, both for the user and the platform (Thomas and Cook, 2005; Isenberg et al., 2017). Visual data exploration often follows Shniderman’s mantra ($i.e.$, "overview first, zoom and filter, then details on demand") (Shneiderman, 1996). The work related to visual prediction, time series visualisation and temporal analytical approaches which matches the keywords of the developed work were

explored.

Recent techniques (Badam et al., 2016; Krause et al., 2016) on visualising the time series data supported with mathematical and statistical metrics enable the user to build reasoning about the considered temporal datasets interactively. Visualisation techniques, highlighting the anomalies and underlying trends correlations, through an undirected interactive search (Sacha et al., 2016) were developed. Moreover, time series visualisation were explored by providing examples of simple charts including stacked graphs, index charts, horizon graphs for visualising time series datasets. The representations of time series data become more contextual with the support of cluster, calendar based and, spiral visualisations (Weber et al., 2011). More detailed and aggregated representations, using multiresolution layouts for handling over plotting in large time series datasets were developed (Fu, 2011; Hao et al., 2011). Moreover they also reviewed the data mining method for classification, pattern exploration, segmentation and representation of time series data. Hochheiser and Shneiderman, invented dynamic query tools for time series dataset interactive explorations with user demand detailing (Hochheiser and Shneiderman, 2004). Chronolenses were proposed for time series data visual exploration and correlation analysis (Zhao et al., 2011a,b). Anomaly detection for modelling multiple time series (Chan and Mohoney, 2005), clustering and classification (Liao, 2005) techniques to identify the similarity of data patterns among time series dataset using weighted dynamic time warping (Jeong et al., 2011), distance metrics and agglomerative clustering have been developed. Inter parameters relationships definition rules are revolutionised by Hetland and Saetrom (Hetland and Saetrom, 2005) with rule mining concept for time series database. The scientific temporal data visualisations are frequently used in support of interactive visual analytics and are well accepted within the disciplines (Andrienko and Andrienko, 2003; Navarra et al., 2020).

Moreover, for understanding the temporal datasets and its trends, predicting future and patterns remains a very challenging task with a few interactive visual models and user explorations behaviour support. Predicting the time series data using statistical methodologies like regression analysis, and computational machine learning approaches like neural networks, multilayer perceptron, fuzzy logic and self organising maps have been successfully applied for the existing studies (Lorenc, 1986; Guilherme, 2007; Bollen et al., 2011; Venugopal et al., 2011). Visual prediction approaches in the act of visually predicting a time series variable by observing the predictions from a computational model, shown alongside with the time series representations for social media and financial datasets were designed. (Hao et al., 2011; Lu et al., 2014; Badam et al., 2016). Furthermore,

interaction techniques with engaging the user in an efficient dialogue in the contribution by people and computers to solve the task together $i.e.$, mixed initiative interaction techniques have also been proposed (Horvitz, 1999, 2007; Endert et al., 2012; Kapoor et al., 2012). Data driven forecasting in visual predictions for time series dataset visualisation with highlighting the sequence and pattern in support of approaches to explore correlations in multivariate spatio-temporal data have been designed by Hao et al. (2011); Malik et al. (2012).

However, the increased usage of the environmental monitoring system and sensors installation on a day to day basis has provided more information in monitoring the current environmental conditions. Sensor networking advancement with quality and quantity for air parameters, has given rise to an increase in techniques and methodologies supporting temporal data interactive visualisation analyses (Hart, 2006; Bogue, 2008). Moreover, there exists a gap between the environment as observed and its digital representation in the user govern time frame for temporal data interactive analysis. Visualisation of meteorological and pollution data history and context plays an essential role in visual data mining, especially in exploring the large and complex datasets and environmental conditions. Including the context and historical information in the visualisation could improve user understanding of the environmental dataset exploration process and enhancing the re-usability of mining and managing techniques and parameters analysis to achieve the required insights. Although, traditional approaches cannot fully support the visual exploration of future trends in complex multivariate time series datasets such as weather, and healthcare, mainly due to their lack of consideration of inter-variable relationships ($e.g.$, if PM_{10} increases, NO_2 decreases). Exploring these relationships through “what if” questions ($e.g.$, what if PM_{10} increases?) could help the user to better judge the future environmental conditions than blindly trusting computational models that lack contextual information. Thus, there is still a gap the user likely needs to bridge for comprehending the situation.

3

Machine Learning Algorithms for Predictions

This chapter discusses the implementations of the first objective of the thesis given in the section 1.5. These findings and developments have passed double peer review during a publication process already.

1. The first development consists of three One Dimensional (1D) algorithms based on LSTM, RF and SVM (section 3.1),

2. The second technique is based on One Dimensional (1D) Convolutional Neural Network (CNN) called as 1DS, followed by the developed third technique of 1D Multiple CNN (1DM) that combines several 1DS but with different views of the same input, therefore, learning more information compared to the 1DS (section 3.2), and

3. The last technique builds upon the 1DM method and develops a multiple CNN architecture with multiple input features, combined

Parts of this chapter have previously been published in:

Harbola, S. and Coors, V. (2019b), 'Comparative analysis of LSTM, RF and SVM Architectures for Predicting Wind Nature for smart city planning', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-4/W9, https://doi.org/10.5194/isprs-annals-IV-4-W9-65-2019, 65–70;

Harbola, S. and Coors, V. (2019a), 'One Dimensional Convolutional Neural Network Architectures for Wind Prediction', *Energy Conversion and Management*, 195, https://doi.org/10.1016/j.enconman.2019.05.007, 70–75;

Harbola, S. and Coors, V. (2019c), 'Convolutional Neural Network Architectures for Wind Analysis,', *EAWE PhD Seminar 2019 29-31 Oct 2019 Nantes, France*, https://eawephd2019.sciencesconf.org/285035,;

Harbola, S. and Coors, V. (2021a), 'Deep learning model for wind forecasting', *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 10, https://doi.org/10.1007/s41064-021-00185-6.

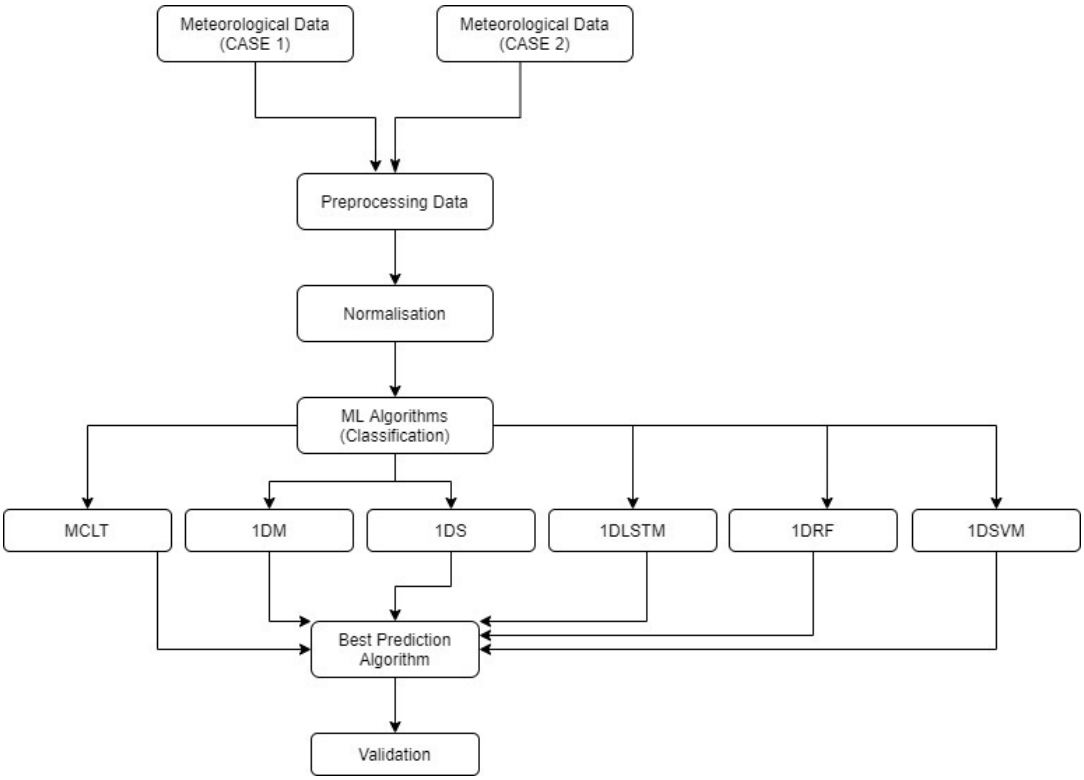


Figure 3.1: Algorithms comparative analysis flow chart.

with multiple LSTM, along with densely connected convolutional layers (section 3.3).

The following chapter explains the experimental setup of the developed prediction models. Afterwards, results and findings are discussed, elaborating the analyses of algorithms, explaining the techniques that are used in this thesis. The chapter summary in section 3.4, explains the advantages and limitations of the developed models.

Developed Methods The developed methods are designed to provide an accurate framework that helps to analyse the meteorological and pollution parameters nature assessments and prediction. In the following sections, the methods are discussed in the order they are developed along with their findings and advantages. The shortcomings of the prior methods are improved in the later designed models. A comparative analysis is put together in investigating the improved accuracy of the developed predicting models. Figure 3.1 provides a brief overview of this chapter workflow along with highlighting the motivation behind this work.

Dataset Used The wind datasets of Stuttgart (Germany) and Netherlands are used in this study. The historical data of 30 years from 1987 to 2017 are taken from Stuttgart (Stadtklima-Stuttgart, 2021), the climate and air measuring station are located in the corner of Hauptstaetter Strasse 70173 Stuttgart, Germany. This dataset contains values of the wind speed and direction at an interval of thirty minutes. The second dataset is from Netherlands from the station 210 Valkenburg with 37 years of historical data from 1981 to 2018 (KNMI, 2020). The dataset of each area is grouped according to the individual month by arranging past data first and most recent data later, thereby helping in predicting the dominating wind speed and direction on a monthly basis. The above datasets are used for all the following designed ML techniques.

3.1 Long Short Term Memory, Random Forest & Support Vector Machine

This study develops three One Dimensional (1D) algorithms based on LSTM, RF and SVM with the following contributions,

1. dominant wind speed and direction predictions using 1D LSTM (1DLSTM), 1D RF (1DRF), and 1D SVM (1DSVM), without applying any smoothening and noise removal techniques,

2. the time frame of prediction is user-defined,

3. using 1DLSTM, 1DRF and 1DSVM as classification instead of regression to enhance accuracy, and

4. comparative study of the 1DLSTM, 1DRF and 1DSVM architectures.

The designed models will provide foreknowledge of wind nature of an area, thereby helping in the proper selection of sites for wind turbine installation. This will provide more utilisation of renewable energy for safe and better city planning, that in turn would help in efficient management and development of the city's resources.

The remaining work is organised as follows, developed methodologies are discussed in subsection 3.1.1, subsection 3.1.2 explains the results and subsection 3.1.3 provides discussion, followed by conclusion in subsection 3.1.4.

Table 3.1: The designed various classes ranges.

Class	Lower limit	Upper limit
1	$\mu - k_1\sigma$	$\mu + k_1\sigma$
2	$\mu + k_1\sigma$	$\mu + k_2\sigma$
3	$\mu + k_2\sigma$	$\mu + k_3\sigma$
4	$\mu + k_3\sigma$	$+\infty$
5	$\mu - k_2\sigma$	$\mu - k_1\sigma$
6	$\mu - k_3\sigma$	$\mu - k_2\sigma$
7	$-\infty$	$\mu - k_3\sigma$

3.1.1 Methodology

The wind dataset comprises wind speed and direction with temporal resolution t and t_i ($i \rightarrow 1$ to k) denotes speed and direction at time i, where 1 and k are the first and last values in the dataset, respectively. Multiple samples are designed using the dataset for training and testing the designed algorithms. A sample consists of a feature vector as an input with a corresponding output class. V_b (a scalar) consecutive values of wind speed from t_i to t_{i+V_b} form a feature vector of dimension $V_b \times 1$ which is the input of the sample. V_f (a scalar) successive values of wind speed after the last value in the input $i.e.$, t_{i+V_b}, are used to define the sample's output class. Mean (μ), and standard deviation (σ) of the wind speed of the entire dataset are calculated. Various class boundaries are designed using μ and σ as shown in Table 3.1. Among V_f, count of values occurring in each class in Table 3.1 is noted, and the class that has a maximum count $i.e.$, dominant, is assigned to the sample. Similarly, multiple samples based on wind speed are created by taking V_b values in the corresponding input from t_i to t_{i+V_b} by varying i from 1 to k - V_f, at an increment of 1. The outputs of these samples are designed as discussed above. Likewise, samples based on wind direction are created where direction instead of speed is considered both in the input and output and μ and σ of direction are calculated. Thus, at this stage, for V_b values in the input from t_i to t_{i+V_b}, there will be two sets of samples, one based on wind speed and other based on wind direction.

The developed 1DLSTM is a special kind of Recurrent Neural Networks (RNN) capable of learning long term dependencies with chain like structure. Figure 3.2 M_{LSTM} model shows the designed architecture for the classification analysis approach for a temporal meteorological dataset. It has an input layer, four neural layers (N1, N2, N3, N4), $i.e.$, three sigmoid layers along with tanh layer and an output layer. The input layer is 1D of the size of V_b. The input layer is successively followed by 1D N1, N2, N3 and N4, with the output layer in the end. The output layer is a softmax

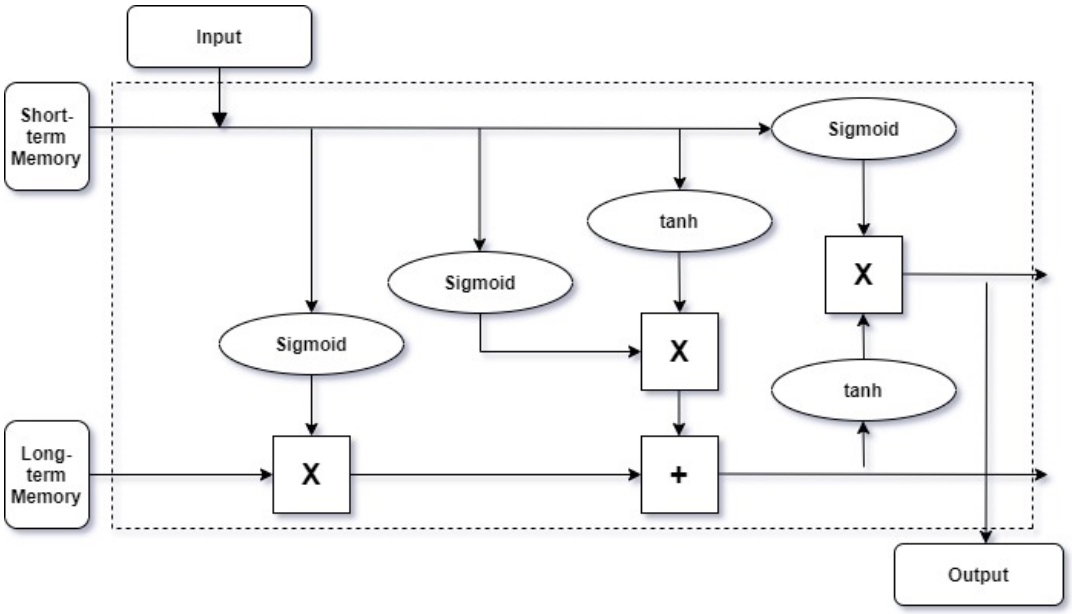


Figure 3.2: 1DLSTM model designed architecture overview.

layer (Goodfellow et al., 2016), having the number of neurons same as the number of the classes. There are seven classes in the present study as shown in Table 3.1.

The designed 1DSVM algorithm classifies the data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for the 1DSVM signifies the one with the largest margin between the two classes. The margin defines the maximal width of the slab parallel to the hyperplane that has no interior data point in time. The support vectors are the data point that are closest to the separating hyperplane; these data points are on the bounds of the slab. 1DSVM can be used when data has exactly two classes. However, multiple classes can be classified using the one-vs-all (OVA) approach, one-vs-one (OVO), and all-vs-all (AVA) approach. In this study, OVO method along with nonlinear Radial Basis Function (RBF) kernel, have been used for classification.

The implemented 1DRF algorithm uses a decision tree as a decision support tool for classification. 1DRF uses a tree like graph to show the possible consequences. When the 1DRF is given a training sample, it formulates a set of rules which are used to perform predictions. Moreover, 1DRF uses sufficient decision trees, to ensure the classifier does not overfit the model while taking the average of all the predictions to remove the biases. The advantage of the 1DRF as a classifier is that it can handle

missing values, and the classifier can be modeled for categorical values and to get the relative feature importance, that contributes in selection of the most favorable features for the classifier. Therefore 1DLSTM, 1DSVM and 1DRF are used to predict wind speed and direction separately. When predicting dominant speed, samples based on speed are used to train and test the 1DSVM. When the dominant direction is to be predicted, then the samples based on direction are used to train and test the 1DSVM. Similarly, the 1DLSTM and 1DRF are trained and tested with samples based on speed for dominant speed prediction and direction for dominant direction prediction. During training, the sample's feature vector of dimension $V_b \times 1$, forms the input of the 1DLSTM, 1DSVM and 1DRF, while the sample's output class forms the output of the 1DLSTM, 1DSVM and 1DRF.

3.1.2 Results

The developed algorithms were implemented using Python and executed with four cores on Intel® Core ™ i7- 4770 CPU @3.40 GHz. Stuttgart's 30 year historical data was separated by month to create monthly data. Similarly, according to each month, 37 years of historical data from the Netherlands was grouped. For each month data (individually for Stuttgart and Netherlands), each algorithm was executed separately to predict the dominant wind speed and direction. Several samples were created from the data of a month, each with input and corresponding output, as described in subsection 3.1.1.

Values of k_1, k_2 and k_3 (Table 3.1) were determined empirically as 0.15, 0.45 and 0.65 for both speed and direction. It ensured that adequate number of samples was present in each class. Also, Synthetic Minority Oversampling Technique (SMOTE) was utilized to do up-sampling of the classes having inadequate number of samples. V_b and V_f were taken as 50 in this study. All the samples for a month, were randomly separated into training and testing samples with 40% of the total samples as testing samples. The designed algorithms were trained and tested with these samples. In order to calculate the average accuracy values, the previous procedure of random division of the total samples into training and testing and the training of the designed methods was repeated ten times, taking into account the randomness of division into training and testing.

1DLSTM learning curves for the testing samples of Stuttgart, September's month are shown in Figure 3.3, where blue curve is for predicting the dominating speed, and orange curve is for predicting dominating direction. Similar learning curves were obtained for the other months as well. Total accuracies of classification for different months for the developed algorithms both for Stuttgart and Netherlands for the wind speed and

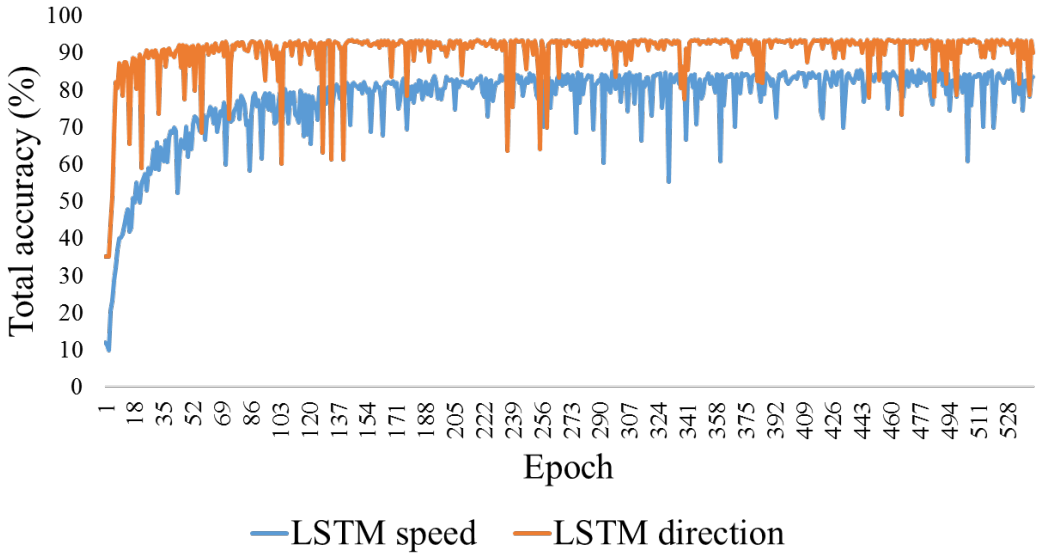


Figure 3.3: Learning curves for testing samples.

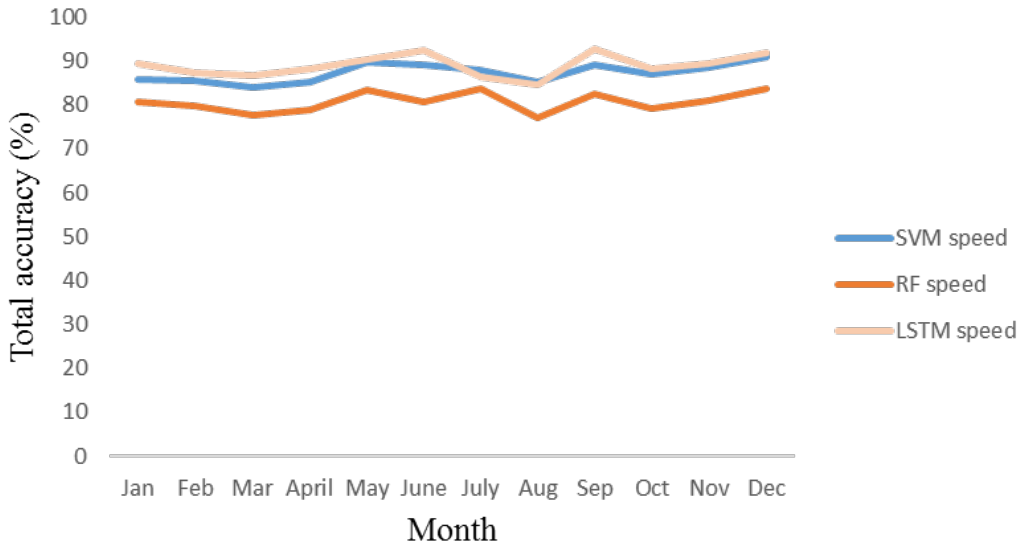


Figure 3.4: Total accuracy for different months for Stuttgart (in speed case).

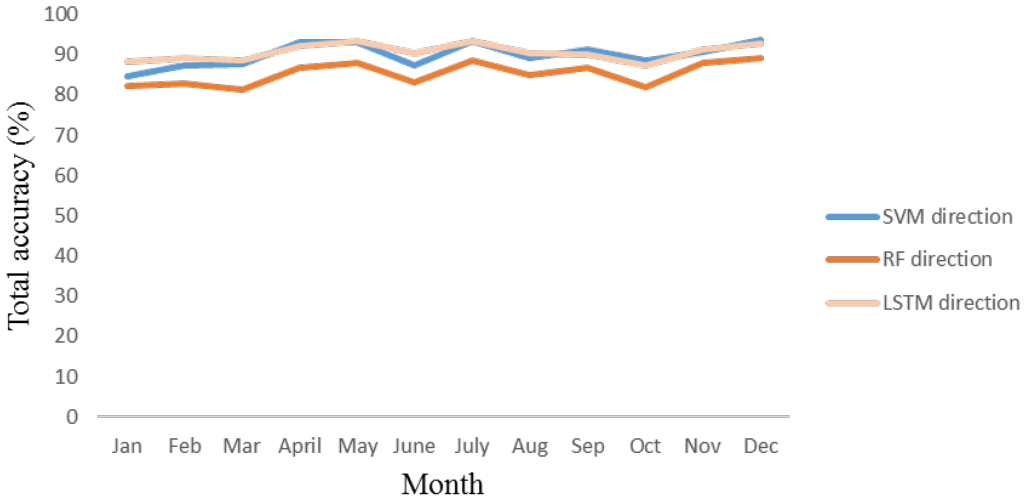


Figure 3.5: Total accuracy for different months for Stuttgart (in direction case).

direction cases (separately) are shown in Figure 3.4, Figure 3.5, Figure 3.6, and Figure 3.7.

For Stuttgart data, minimum and maximum total accuracy for predicting dominant speed are 77.2% and 87.3% respectively using the designed 1DRF method, 83.9% and 90.9% respectively using the developed 1DSVM method, and 84.5% and 92.4% respectively using the developed 1DLSTM method. Prediction of dominant direction of Stuttgart data, using the 1DRF results in minimum and maximum total accuracy of 81.6% and 89.3% respectively, 84.6% and 93.8% respectively using the 1DSVM and 87.4% and 93.3% respectively using the 1DLSTM (Figure 3.4 and Figure 3.5). Similarly, for Netherlands data, minimum and maximum total accuracy for predicting dominant speed are 80.7% and 88.7% respectively using the 1DRF method, while 86.9% and 92.8% respectively using the designed 1DSVM method, and 87.7% and 93.9% respectively using the developed 1DLSTM method. Prediction of dominant direction using the 1DRF method results in minimum and maximum total accuracy of 80.6% and 88.3% respectively, whereas 87.7% and 92.4% respectively using the 1DSVM method and 87.2% and 94.7% respectively using the 1DLSTM method, for the same Netherlands data (Figure 3.6 and Figure 3.7).

Figure 3.8 and Figure 3.9 show precision and recall values of dominant speed and direction prediction for June month of Stuttgart. Similar results were obtained for other months as well. Maximum precision and recall values are 92.6% and 92.7% respectively using 1DRF, 92.4% and 93.1% re-

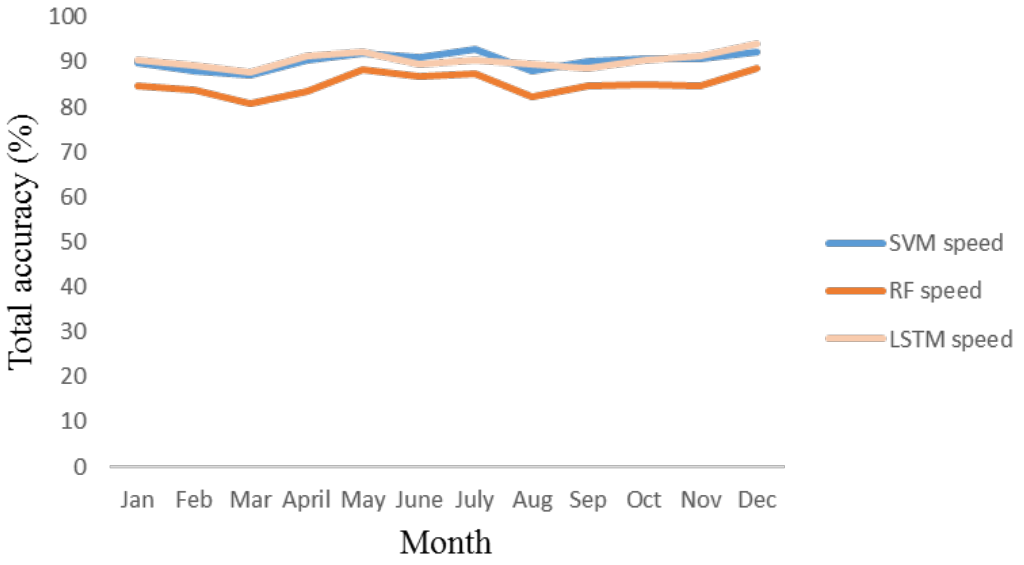


Figure 3.6: Total accuracy for different months for Netherlands (in speed case).

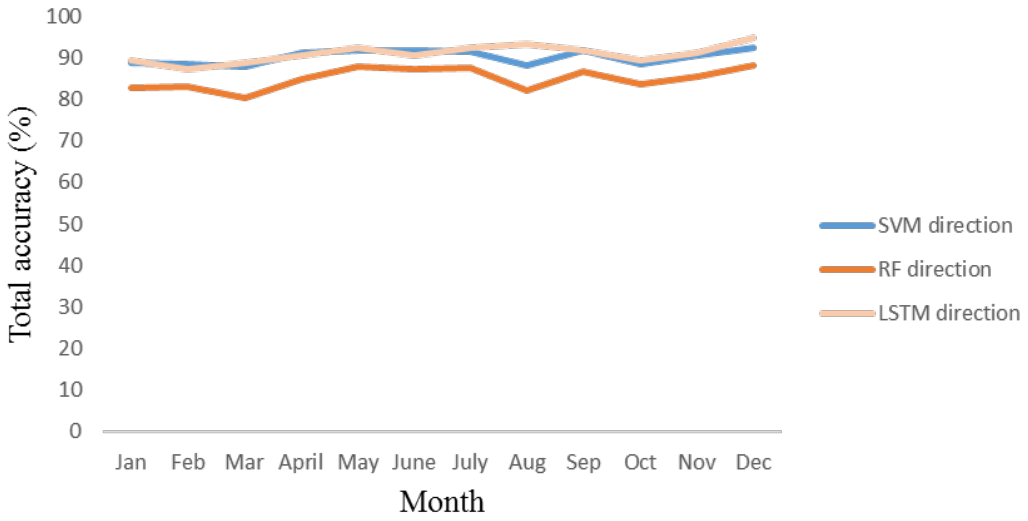


Figure 3.7: Total accuracy for different months for Netherlands (in direction case).

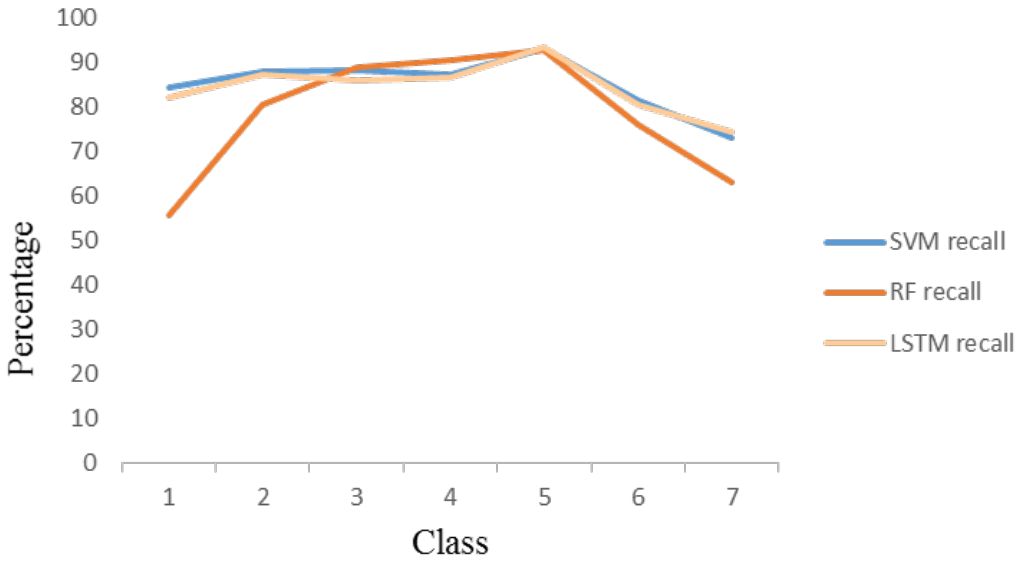


Figure 3.8: Precision values of different classes for June month of Stuttgart.

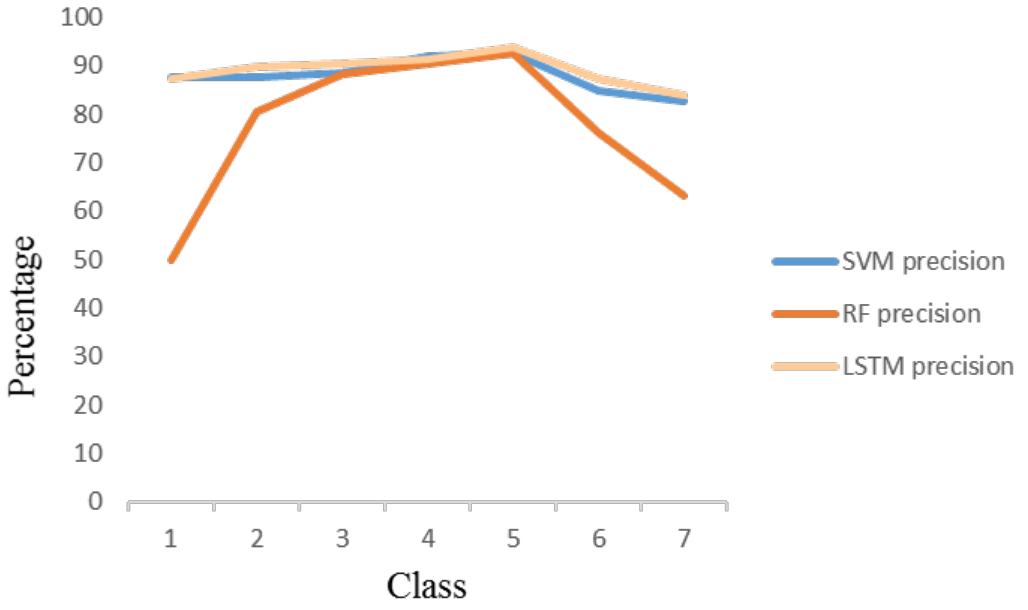


Figure 3.9: Recall values of different classes for June month of Stuttgart.

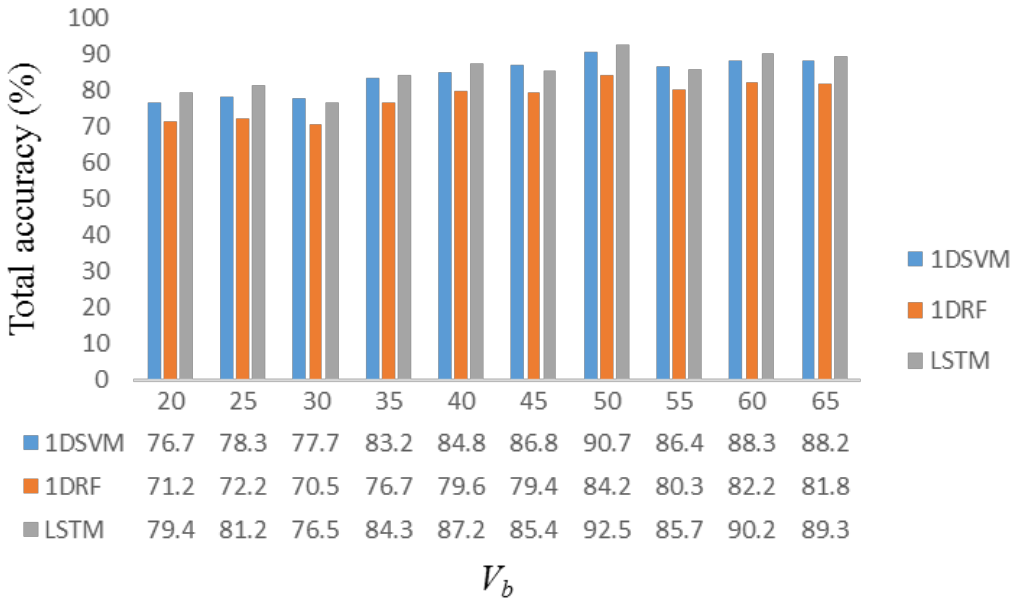


Figure 3.10: Total accuracy variation for different V_b, with $V_f = V_b$.

spectively using 1DSVM and 93.7% and 93.4% respectively using 1DLSTM. Further, value of V_b was varied to carry out its sensitivity analysis. In first case, V_b was varied and V_f remains same as V_b. In second case, V_f was kept constant at 50 while V_b varies. These were performed for June month of Stuttgart and total accuracy for dominant speed prediction is shown in Figure 3.10 and Figure 3.11. As value of V_b increases from 20 to 50, the total accuracy increases, which after 50, remains approximately similar for the 1DLSTM, 1DRF and 1DSVM. Thus, V_b and V_f were taken as 50 in this study. Higher V_b value presents a larger feature vector as input of a sample which contributes to more information and 1DLSTM, 1DRF and 1DSVM perform better.

3.1.3 Discussion

The designed 1DRF using multiple decision trees is able to detect patterns in the input feature vector of a sample and is able to predict dominant wind speed and direction with good accuracy. The 1DSVM method maximises the margin between the support vectors and the hyperplane and is able to perform better using a non linear Radial Basis Function (RBF) kernel. The 1DSVM performs better by up to 8.4% and 6.4% in comparison to 1DRF for predicting dominant wind speed and direction respectively. The 1DLSTM

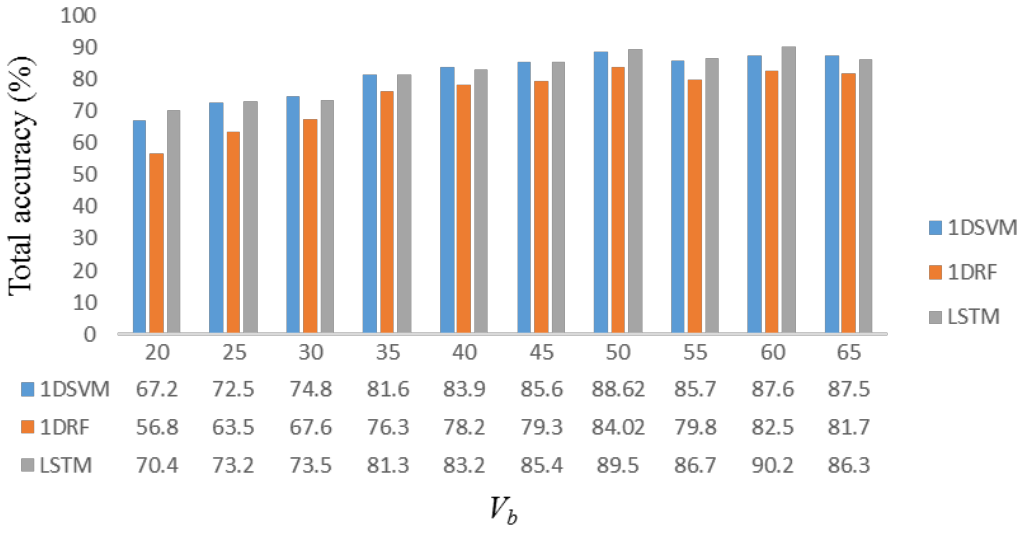


Figure 3.11: Total accuracy variation for different V_b, with $V_f = 50$.

with four recurrent layers with long term dependencies performs better by up to 9.3% and 7.9% in comparison to 1DRF for predicting dominant wind speed and direction respectively, and by up to 1.0% and 1.5% in comparison to the 1DSVM. Moreover, 1DLSTM and 1DSVM share similar results for most of the dataset cases. The performance of 1DLSTM can be improved with increased numbers of neural layers with advanced activation functions, though it requires better hardware resources.

The higher number of classes (7) in the output ensures that the designed methods are able to learn varieties of samples during the training and can predict with good accuracies during the testing. Moreover, the higher number of classes helps to identify the sudden changes in the wind speed and direction and ensures that most of the minor and major details are learnt during the training phase. During the designing of the samples, their output classes were decided statistically using μ and σ of a particular month's wind dataset, thereby representing the dataset better. However, currently with 7 classes in the 1DLSTM, 1DRF and 1DSVM, for Stuttgart, total accuracy enhances by up to 6.7% and 6.9% (for May month) using 1DLSTM and 6.3% and 6.6% (for May month) using 1DSVM for wind speed and direction, respectively, with respect to the corresponding month using 1DRF (Figure 3.4 and Figure 3.5). Similarly, for Netherlands, using the 1DLSTM, total accuracy is enhanced by up to 7.1% and 5.4% (for March month) and using 1DSVM by 6.9% and 5.5% (for October month) for wind speed and direction, respectively, with reference to the corresponding

month using 1DRF (Figure 3.6 and Figure 3.7). The input feature vectors of samples are based on the original wind data values. The developed 1DLSTM, 1DRF and 1DSVM algorithms take as input the original data without applying any smoothening technique to filter out the noise and have only a single user-defined parameter V_b, thus making these algorithms less susceptible to the noise along with the use of real data and minimum parameter tuning. The comparative study of the designed 1DSVM with De Giorgi et al. (2014) and Yuan et al. (2015) is performed, along with the comparison of 1DLSTM with Ghaderi et al. (2017), as these algorithms are nearest to the developed methods. De Giorgi et al. (2014) and Yuan et al. (2015) have used SVM and LSTM but with regression analysis (LSSVM) and smoothening and filtering techniques have been applied to remove noise from the dataset. Similarly, Ghaderi et al. (2017) have used LSTM with regression and noise has been removed from the dataset by smoothening and filtering, thereby modifying the originality of wind dataset. The samples used in the present study are utilized to train and test the De Giorgi et al. (2014); Yuan et al. (2015), and Ghaderi et al. (2017) architectures. In this case outputs of the samples are changed to real values ($i.e.$, regression) unlike classification as in the 1DLSTM, 1DSVM and 1DRF. Values of V_b and V_f are kept as same. Symmetric Mean Absolute Percentage Error (SMAPE) (Shuyang et al., 2017) for wind speed using De Giorgi et al. (2014), Yuan et al. (2015), and Ghaderi et al. (2017) are more or less similar and is 18.2% for V_f = 15 and increases as V_f increases, reaching up to 32.5% for V_f = 50. Likewise results using De Giorgi et al. (2014), Yuan et al. (2015), and Ghaderi et al. (2017) architectures were obtained for the wind direction. Thus, error increases substantially when more values are predicted in future using state-of-the-art SVM and LSTM based regression architectures De Giorgi et al. (2014), Yuan et al. (2015), and Ghaderi et al. (2017). However, the designed 1DSVM method for predicting dominant speed and direction based on classification, achieves high accuracy reaching up to 93.9% and 94.7% (1DLSTM), up to 92.8% and 93.8% (1DSVM) and up to 88.7% and 89.3% (1DRF) for speed and direction, respectively for V_f = 50 even without applying any smoothening or filtering to the original data. Thus, the implemented 1DLSTM, 1DSVM and 1DRF methods are suited for predicting dominant speed and direction for a larger time period in the future unlike the De Giorgi et al. (2014), Yuan et al. (2015), and Ghaderi et al. (2017) regression based architectures.

3.1.4 Conclusion

The integration of new knowledge, innovative technologies in sustainable transformation is a motive of this work. The algorithms using 1DLSTM,

1DRF and 1DSVM have been developed for predicting the dominant wind speed and direction classes. V_b continuous values of the wind speed and direction separately form a sample's input and predict the dominating speed and direction, among V_f values after the last value in the sample input, using 1DLSTM, 1DRF and 1DSVM. The developed algorithms show promising results when trained and tested using wind datasets of Stuttgart and Netherlands. The maximum total accuracy using the 1DRF in case of Stuttgart for predicting dominant speed and direction are 83.7%, 89.3% respectively, and for Netherlands 88.7%, 88.3% respectively. Meanwhile, using the 1DSVM maximum total accuracy for predicting the dominant speed and direction for Stuttgart are 90.9%, 93.8% respectively, and for Netherlands 92.8%, 92.4% respectively. Further, using the 1DLSTM maximum total accuracy for predicting the dominant speed and direction for Stuttgart are 92.7%, 93.5% respectively, and for Netherlands 93.9%, 94.7% respectively. The total accuracy enhances by up to 6.7% and 6.9% (for May month) using 1DLSTM and 6.3% and 6.6% (for May month) using 1DSVM for Stuttgart's wind speed and direction, respectively, with respect to the corresponding month using 1DRF. At the same time for Netherlands, total accuracy using the 1DLSTM is enhanced by up to 7.1% and 5.4% (for March month) and using 1DSVM by 6.9% and 5.5% (for October month) for wind speed and direction, respectively, with reference to the corresponding month using 1DRF. The advantage of these methods is that they do not apply any smoothening and noise removal techniques and are based on classification approach. LSTM learns long term dependencies in the temporal data, SVM finds the probable hyperplane between points of different classes and RF uses multiple decision trees. However, in these algorithms a limited number of features and classes are used. A better approach is required that could incorporate multiple features and more number of classes.

3.2 One Dimensional Convolutional Neural Network Architectures

This study develops

1. 1D Single CNN (1DS) for predicting dominant wind speed and direction of 1D time series wind data,

2. 1D Multiple CNN (1DM) with multiple views of time series wind data to enhance the 1DS accuracy, and

3. the 1DS and 1DM as classification methods working on the original 1D wind data values and future time frame of prediction depends on the user.

The rest of the sections are organised as follows: subsection 3.2.1 explains the developed algorithms, section 3 describes the datasets used in this study, followed by subsection 3.2.2 and subsection 3.2.3 where detailed results are discussed and the conclusion is given in subsection 3.2.4.

3.2.1 Methodology

The wind dataset has values of wind speed and wind direction at regular time interval t. t_j ($j \rightarrow 1$ to n), gives values of wind speed and direction at time j, where 1 and n are the first and last values, respectively, in the dataset. A sample has multiple input values and an output class. The sample comprises, say W_S (a scalar), consecutive values from t_j to t_{j+W_S} of the dataset with two features of speed and direction, in the input. W_B (a scalar) successive values in the dataset, after the last value in the sample's input $i.e.$, t_{j+W_S}, are used to define the output class of the sample separately for speed and direction. For this, mean (μ) and standard deviation (σ) of the speed and direction for the complete dataset are computed separately. Table 3.2 lists the lower (inclusive) and upper (exclusive) boundary ranges for the various classes based on μ and σ.

In this study, same boundary ranges are used for defining the output class based on speed and direction. Among W_B, respective count of the values of the speed occurring in these classes (Table 3.2) are calculated and the class with the maximum count, $i.e.$, dominant, is assigned to the sample. Similarly, among W_B, count of the values of the direction occurring in each class is computed and the class with the maximum count, $i.e.$, dominant, is assigned to the sample. Thus, the sample at this stage has same W_S values with two features of speed and direction in the input, but two output classes, one based on the speed and the other based on the direction. Moreover, scalar W_B multiplied by the temporal resolution (t) of the wind dataset, gives the future prediction time frame for the dominating wind speed and wind direction. Likewise, more such samples are designed by taking W_S values in the input from t_j to t_{j+W_S} of the dataset by varying j from 1 to ($n - W_B$), at an increment of 1. The output classes of these samples are designed accordingly as discussed above.

The developed 1DS has an input layer, three convolutional layers (C_1, C_2, C_3), two fully connected layers (F_1, F_2) and an output layer (Figure 3.12). The input layer is 1D of size of W_S. A sample which is passed through the 1DS input layer, comprises W_S consecutive values from the dataset with

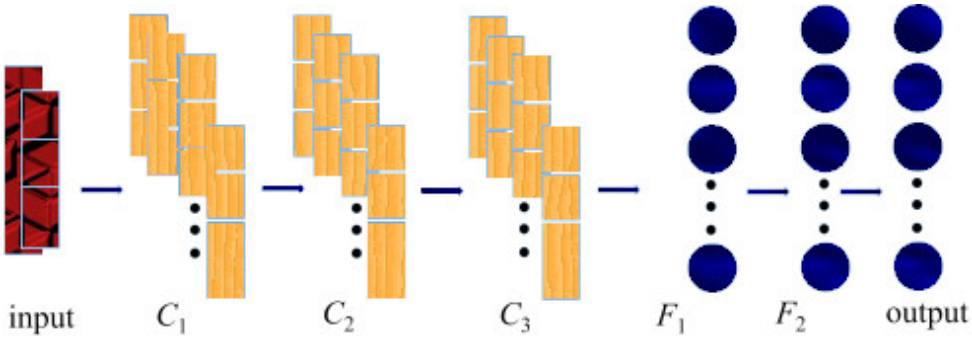


Figure 3.12: Single CNN (1DS) architecture.

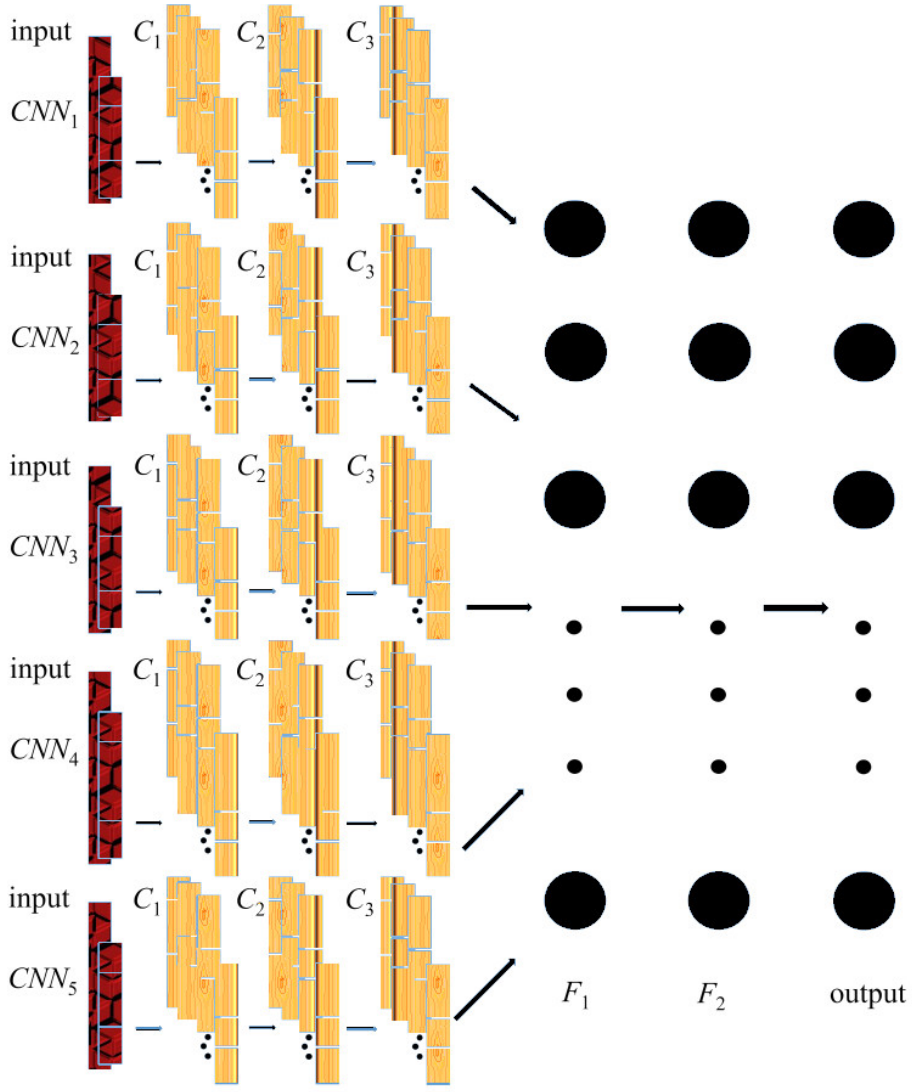


Figure 3.13: Multiple CNN (1DM) architecture.

Table 3.2: The designed various classes ranges.

Class	Lower limit	Upper limit
1	$\mu - k_1\sigma$	$\mu + k_1\sigma$
2	$\mu + k_1\sigma$	$\mu + k_2\sigma$
3	$\mu + k_2\sigma$	$\mu + k_3\sigma$
4	$\mu + k_3\sigma$	$\mu + k_4\sigma$
5	$\mu + k_4\sigma$	$\mu + k_5\sigma$
6	$\mu + k_5\sigma$	$+\infty$
7	$\mu - k_2\sigma$	$\mu - k_1\sigma$
8	$\mu - k_3\sigma$	$\mu - k_2\sigma$
9	$\mu - k_4\sigma$	$\mu - k_3\sigma$
10	$\mu - k_5\sigma$	$\mu - k_4\sigma$
11	$-\infty$	$\mu - k_5\sigma$

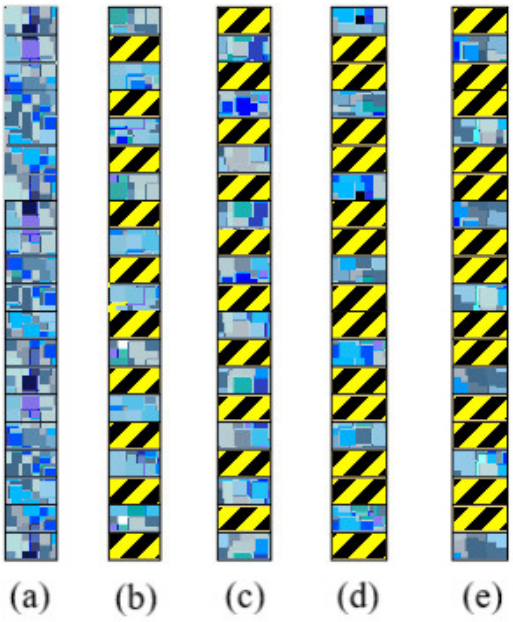


Figure 3.14: Inputs to 1DM. (a) represents W_S values of a sample. (a), (b), (c), (d) and (e) represent input to CNN_1, CNN_2, CNN_3, CNN_4, CNN_5, respectively where blue-squares denote values included in input and yellow-black strips denote excluded values.

two features of speed and direction (as discussed above). The input layer is successively followed by 1D C_1, C_2, C_3 with C_3 connected to F_1 and F_1 is followed by F_2, with the output layer in the end. The output layer is a softmax layer (Memisevic et al., 2010; Su et al., 2015), with the number of neurons same as the number of the classes. There are eleven classes in the present study as shown in Table 3.2. The 1DS is trained and tested separately for the prediction of the dominant speed and direction. When the 1DS is used for dominant speed prediction, then the samples output classes based on the speed are considered. When the 1DS is used for dominant direction prediction, then the samples output classes based on the direction are considered. Thus, the input values of the samples remain the same, but the corresponding output classes change based on the speed or direction prediction. Therefore, for a test sample, the 1DS can predict the speed and direction classes separately. The developed 1DM has five single CNN, say (CNN_1, CNN_2, CNN_3, CNN_4, CNN_5), as shown in Figure 3.13. Each of these CNN_i ($i \rightarrow 1$ to 5) has its own input layer, C_1, C_2, C_3, as in the 1DS and C_3 of each CNN_i connects to the common F_1 which is followed by F_2 and the softmax layer. The number of neurons in the softmax layer of the 1DM, are same as in the 1DS. In Figure 3.14, the input of CNN_1 is same as input of the 1DS with W_S values and two features. The input of CNN_2 takes W_S values but at an increment of two, starting from the first value, thus, the input has half of the W_S values. Similarly, the input of CNN_3 takes W_S values at an increment of two but starts from the second value. The input of CNN_4 and CNN_5 are designed similarly by using an increment of three, starting from the first and the second values of W_S respectively as shown in Figure 3.14. Thus, the sample having W_S values is passed through the five different CNNs of the 1DM by taking the corresponding inputs. Moreover, as in the 1DS, for the designed 1DM also the input values of the samples remain the same, but the corresponding output classes change based on the speed or direction prediction.

3.2.2 Results

The developed algorithms were implemented using Keras library (Chollet, 2017) with TensorFlow in backend in Python and executed on Intel® Core ™ i7- 4770 CPU @3.40 GHz having four cores. The 30 years historical data from Stuttgart was separated by each month to create respective months data. Similarly, 37 years historical data from Netherlands was grouped according to each month. Each algorithm was executed separately for each month data (of Stuttgart and Netherlands individually) for predicting the dominating wind speed and direction. Several samples, each having input and corresponding output, were created from a month's data as

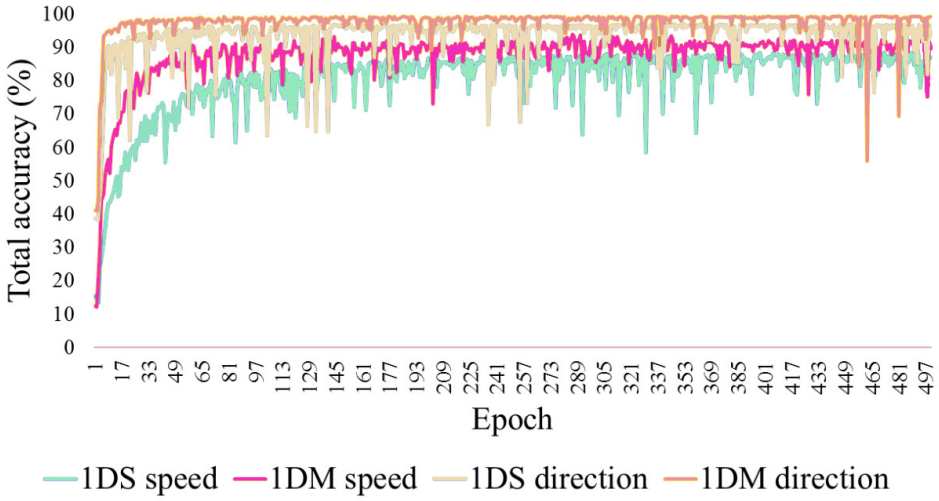


Figure 3.15: Learning curves for testing samples.

described in subsection 3.2.1. Values of k_1, k_2, k_3, k_4 and k_5 (Table 3.2) were taken as 0.15, 0.45, 0.65, 0.95 and 1.25 respectively (same for both speed and direction), so that a sufficient number of samples occur in each class. Moreover, Synthetic Minority Oversampling Technique (SMOTE) was used to do up-sampling of the classes having less number of samples. W_S and W_B were taken as 50. Total samples for a given month were randomly split into training and testing with 35% of the total samples as the testing samples. The developed algorithms were trained and tested on these samples.

Previous procedure of randomly splitting the total samples into training and testing along with the training and testing of the developed methods was repeated ten times to calculate the average accuracies values, considering the randomness of splitting into training and testing. The developed algorithms use Exponential Linear Units (ELUs) (Clevert et al., 2016; Pedamonti, 2018) as activation function with α of 3.0, kernel size of 3 and stride of 1 for all the convolutional layers and these values have been selected empirically. The batch normalisation is used after every convolution layer (Jung et al., 2019) and dropout of 0.20 is used, which along with ELUs prevent overfitting ($i.e.$, network shows high accuracy during training but less accuracy when new data is given during testing). The number of feature maps in C_1, C_2 and C_3 of 1DS are 10, 10, and 20, respectively, whereas the number of neurons in F_1 and F_2 of 1DS are 200 and 100 respectively. Further, the number of feature maps in C_1, C_2 and C_3 of each of CNN_i are 10, 10, and 20, respectively, whereas the number of neurons in F_1 and F_2 of 1DM are same as in 1DS.

Learning curves for the testing samples of May month of Stuttgart are

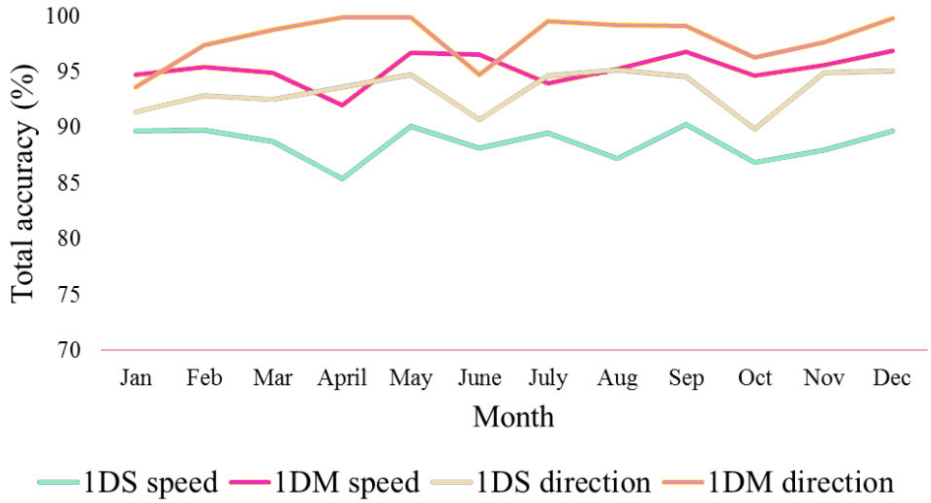


Figure 3.16: Total accuracy for different months for Stuttgart.

shown in Figure 3.15, where cyan and pink color curves are of the 1DS and 1DM respectively for predicting dominating speed and off-white and brown color curves are for the 1DS and 1DM respectively for predicting dominating direction. Similar learning curves were obtained for the other months as well. Classification accuracies in terms of the total accuracies for the designed methods for different months are shown in Figure 3.16, Figure 3.17. Minimum and maximum total accuracy using the 1DS for predicting dominant speed are 85.4% and 90.2% respectively, whereas using the 1DM these are 92.0% and 96.8% respectively, for Stuttgart.

Further, minimum and maximum total accuracy using the 1DS for predicting dominant direction are 89.8% and 95.1% respectively, whereas using the 1DM these are 93.6% and 99.7% respectively, for Stuttgart (Figure 3.16). Similarly, for Netherlands, minimum and maximum total accuracy using the 1DS for predicting dominant speed are 90.0% and 95.2% respectively, whereas using the 1DM these are 97.5% and 98.8% respectively and minimum and maximum total accuracy using the 1DS for predicting dominant direction are 91.3% and 94.7% respectively, whereas using the 1DM these are 97.6% and 99.4% respectively (Figure 3.17).

Precision and recall values for predicting dominant speed for various classes for May month (of Stuttgart) are represented in Figure 3.18, and similar were the results for other months. The 1DM has precision and recall values for all the classes above 79.1% whereas the 1DS has above 59.0%. Moreover, sensitivity analysis of W_S was performed by calculating the total accuracy for different W_S, firstly with $W_B = W_S$ (Figure 3.19) and secondly by keeping W_B constant as 50 (Figure 3.20), for May month (of

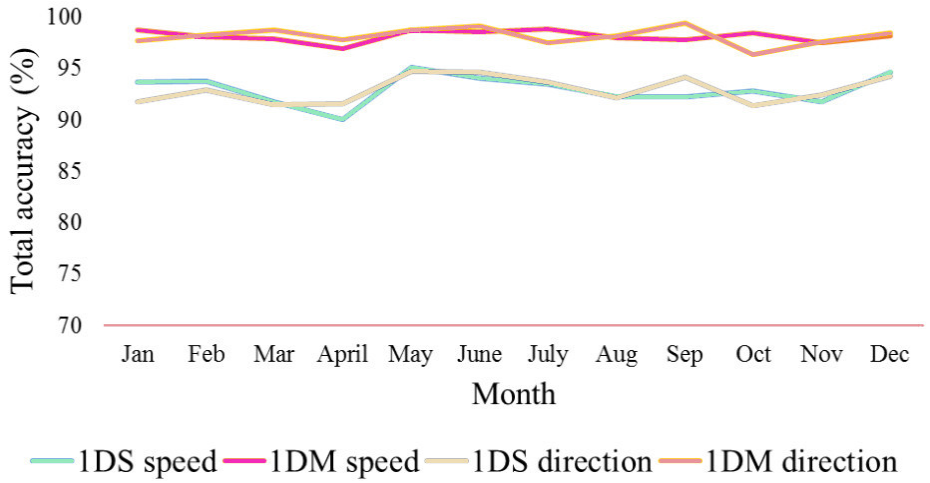


Figure 3.17: Total accuracy for different months for Netherlands.

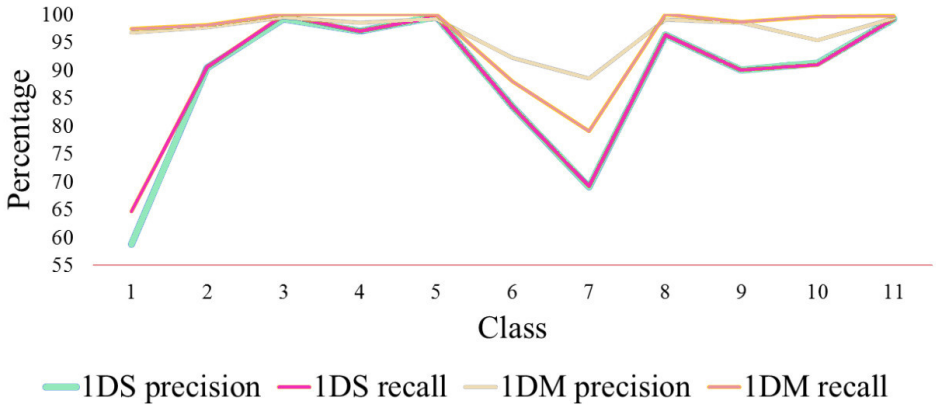


Figure 3.18: Precision and recall values of different classes for May month of Stuttgart.

Stuttgart) to know its effect. In Figure 3.19, Figure 3.20, the total accuracy increases as W_S increases from 20 to 50 and after 50 remains more or less similar for both the 1DS and 1DM when predicting wind speed for May month. Therefore, in this study W_S was taken as 50. The increase in accuracy can be attributed to the more input values in a sample, therefore, more information is passed through the CNNs input layer and the networks perform better.

3.2.3 Discussion

The developed 1DS has three 1D convolutional layers and three fully connected layers (including output layer), where the convolution layers act as

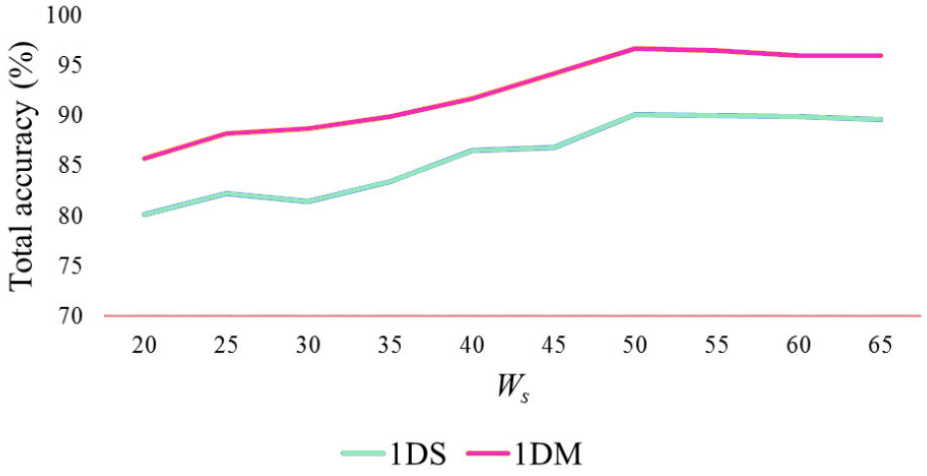


Figure 3.19: Total accuracy variation for different W_S, with $W_B = W_S$.

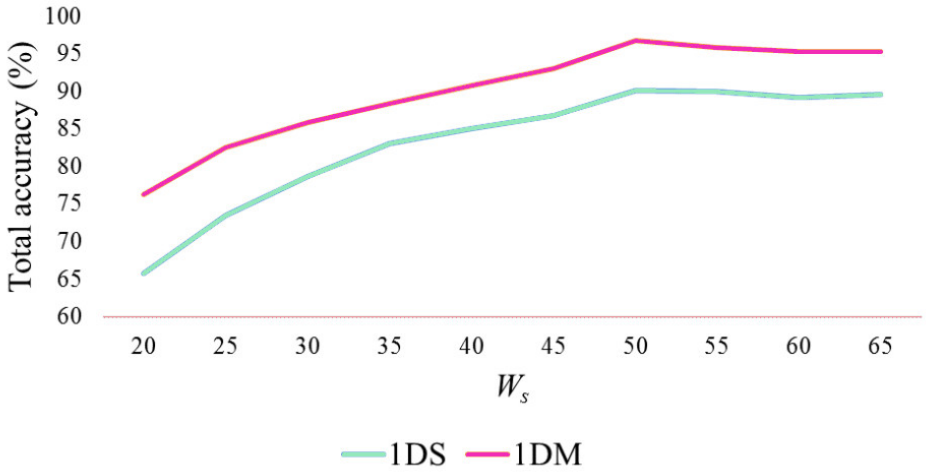


Figure 3.20: Total accuracy variation for different W_S, with $W_B = 50$.

feature extractors from the input wind dataset whereas the fully connected layers act as a classifier thereby predicting the dominant wind speed and direction. An input sample comprising W_S consecutive values from the dataset with two features of speed and direction provide temporal information for both speed and direction and the 1D convolutional operations are able to detect temporal trends and features. The CNN can detect more details with a large number of feature maps (Qi et al., 2016) but due to the hardware constraints in the current study, limited feature maps and neurons have been used. The higher number of classes (11) in the output ensures that the developed methods are able to learn varieties of samples during the training and can predict with good accuracies during the testing. Moreover, the higher number of classes helps to identify the sudden changes in the wind speed and direction and ensures that most of the minor and major details are learnt during the training phase. During the designing of the samples, their output classes were decided statistically using μ and σ of a particular month's wind dataset, thereby representing the dataset better.

The 1DM combines multiple 1DS information by merging last convolutional layer of each 1DS at common F_1. Different variations of W_S in the corresponding input layers of CNN_i ($i \rightarrow$ 1 to 5), provide additional views in terms of temporal resolution of the same input (Figure 3.14), thereby the 1DM learns more information than the 1DS. More views can be used in the 1DM for enhanced accuracy. However, currently with 5 views in the 1DM, for Stuttgart, total accuracy is enhanced by up to 8.4% (for June) and 6.3% (for March) for the wind speed and direction, respectively, with respect to the 1DS (Figure 3.3). Similarly, for Netherlands, using the 1DM, total accuracy is enhanced by up to 6.9% (for April) and 7.3% (for March) for wind speed and direction, respectively, with respect to the 1DS (Figure 3.17). The developed 1DS and 1DM algorithms take as input the original data without applying any smoothening technique to filter out the noise and have only a single user-defined parameter W_S, thus making these algorithms less susceptible to the noise along with the use of real data and minimum parameter tuning. The comparative analysis with the existing literature method Liu et al. (2018) using 1D CNN which is nearest to the designed algorithms is carried out.

1D CNN with the regression concept has been used in Liu et al. (2018) along with the smoothening and filtering of the values of the samples which amends the originality of the wind dataset. When the same samples, comprising W_S (= 50) input values, that are utilised for the developed 1DS and 1DM, are used to train and test the regression CNN architecture Liu et al. (2018) without applying smoothening and filtering, Symmetric Mean Absolute Percentage Error (SMAPE) (Flores, 1986) for wind speed is

14.5% for W_B = 10 and increases as W_B increases, reaching up to 19.5% for W_B = 50. Similar results were obtained for the wind direction. It may be noted that that in this case outputs of the samples are based on real values ($i.e.$, regression) unlike classification as in the 1DS and 1DM. This indicates that error increases significantly as more values in future are predicted using state-of-the-art CNN based regression architecture Liu et al. (2018), whereas, the developed CNN architectures based on classification for predicting dominant speed and direction, give high accuracy reaching up to 99.7% for W_B = 50 even without applying any smoothening to the original data.

This makes the developed 1DS and 1DM architectures suitable for predicting dominant speed and direction for a larger time frame in the future unlike Liu et al. (2018). The accuracies of the designed methods can be enhanced by increasing the number of convolutional and fully connected layers along with more feature maps in convolutional layers. However, it requires better hardware resources and faster graphical processing units (GPUs) as more calculations are to be done during feedforward and backpropagation stages.

3.2.4 Conclusion

In this work, two deep learning algorithms using 1D single CNN and 1D multiple CNN have been designed for predicting the dominant wind speed and direction classes. The 1DS takes W_S continuous values of the wind speed and direction as an input sample and predicts the dominating speed and direction, separately, among W_B values after the last value in the input sample. The designed 1DM combines several 1DS with different views of the same input W_S, thereby learning additional information. The algorithms are trained and tested on the wind datasets of Stuttgart and Netherlands and have shown promising results. Maximum total accuracy using the 1DS for predicting dominant speed and direction are 90.2%, 95.1% respectively, for Stuttgart, and 95.2%, 94.7% respectively, for Netherlands. Maximum total accuracy using the 1DM with 5 views, for predicting the dominant speed and direction are 96.8%, 99.7% respectively, for Stuttgart, and 98.8%, 99.4% respectively, for Netherlands. The 1DM enhances total accuracy by up to 8.4% (for June) and 6.3% (for March) for wind speed and direction, respectively, for Stuttgart, with respect to the corresponding 1DS. Similarly, for Netherlands, total accuracy using the 1DM is enhanced by up to 6.9% (for April) and 7.3% (for March) for wind speed and direction, respectively, with respect to the corresponding 1DS. Further, the 1DM has better performance than the 1DS due to the use of multiple 1DS. In these approaches limited (only two) features based on wind speed and direction

are used in the input layers. Further, the fully connected layers do not have memory to retain the features learnt by neurons from the previous training iterations. Thus, an algorithm is desired that could take multiple features in the input layers as well.

3.3 Multiple Densely Connected Convolutional Neural Network

The present work improves upon the 1DM model (subsection 3.2.1) and develops deep multiple CNN architecture with multiple input features, along with multiple Long Short Term Memory (LSTM) and having densely connected convolutional layers. More number of features in CNN architecture helps in learning the various properties of a sample from finer to coarser levels. Therefore, a large number of features are used in this study. This architecture is called Multiple features, Multiple Densely Connected Convolutional Neural Network ensembles with Multiple LSTM Architecture $i.e.$, MCLT with the following essential contributions,

1. multiple features (58 in total) are used in the input layers for better representation of the temporal wind dataset,

2. fully connected layers are replaced with LSTM layers to provide memory for a longer period and thus improved training of the model,

3. connecting convolutional layers as in the 2D ResNet (for images) architecture are used so that each convolutional layer learns features of previous convolutional layers as well,

4. a higher number of classes (21) are used for analysing detailed trend of the temporal wind dataset, and

5. visual validation of the model's output using wind rose plots.

The author is unable to find any existing literature that has used these five contributions for in depth analysis and prediction of wind nature. The remaining work is arranged as follows: subsection 3.3.1 describes the MCLT architecture followed by section 3 which gives detail of the wind datasets used in the experiments. The subsection 3.3.2 and subsection 3.3.3 present the results and discussion followed by conclusion and recommendations in subsection 3.3.4.

3.3.1 Methodology

The developed MCLT architecture is an advanced deep learning architecture, which is a combination of multiple features, multiple LSTM, and densely connected convolutional layers in the multiple CNN model for the wind nature analysis. The designed multiple features, a total of 58 features, are based on the various combinations of two important temporal wind properties, $i.e.$, wind speed and direction. This ensures that several details of the wind features are learnt by the MCLT. The following sections discuss the design of these multiple features, along with the MCLT framework.

Designing Multiple Features

Wind speed and direction are two input features to the developed architecture. Besides these two features, 56 additional features also form part of the input. Suppose, matrix $M_{i,\ j}$ has r rows and 58 columns, where r equals to the number of temporal wind values present in the dataset (each row of $M_{i,\ j}$ is a time instance for wind dataset), and i, j denote row and column number of a cell respectively, in the matrix. Moreover, each column denotes a feature. The first feature (first column), second feature (second column) comprise the wind speed and direction values, respectively. $M_{i,\ j=3}$ (third feature) is the percentage difference (per) between $M_{i,\ j=1}$ (speed values) and $M_{i-1,\ j=1}$. $M_{i,\ j=4}$ (fourth feature) is the percentage difference between $M_{i,\ j=1}$ and $M_{i-2,\ j=1}$. Similarly, the features from $M_{i,\ j=5}$ to $M_{i,\ j=58}$ are based on the percentage difference (per), standard deviation (std), correlation coefficient ($corcoef$), eigenvalues ($eig1$, $eig2$) and entropy ($entr$) of wind speed and direction. These are discussed in detail in Figure 3.21, where values up to $M_{i-7,\ j}$ are used only due to hardware constraints in the present study, it could be decreased or increased as per available hardware. In Figure 3.21, for example std ($M_{i,\ j=2}$, $M_{i-1,\ j=2}$, $M_{i-2,\ j=2}$) means standard deviation of three quantities inside the brackets. Similar is the explanation of other features in Figure 3.21.

Samples for training and testing the designed architecture are designed using $M_{i,\ j}$. A sample comprises input values and a corresponding output value. Rows from i to i + K_B (and all columns of these rows) of $M_{i,\ j}$ form the input of the sample, where K_B is a scalar. These columns are treated as separate features, each of one dimension, in the input layers of the MCLT as discussed in the next section. The output of the sample is designed using values of speed from $M_{i+K_B+1,\ j=1}$ to $M_{i+K_B+K_F,\ j=1}$, where K_F is a scalar. For this, mean (μ) and standard deviation (σ) of a historical temporal wind dataset are calculated, separately for speed and direction. Then 21 classes are made using μ and σ, as shown in Table 3.3. The μ and σ

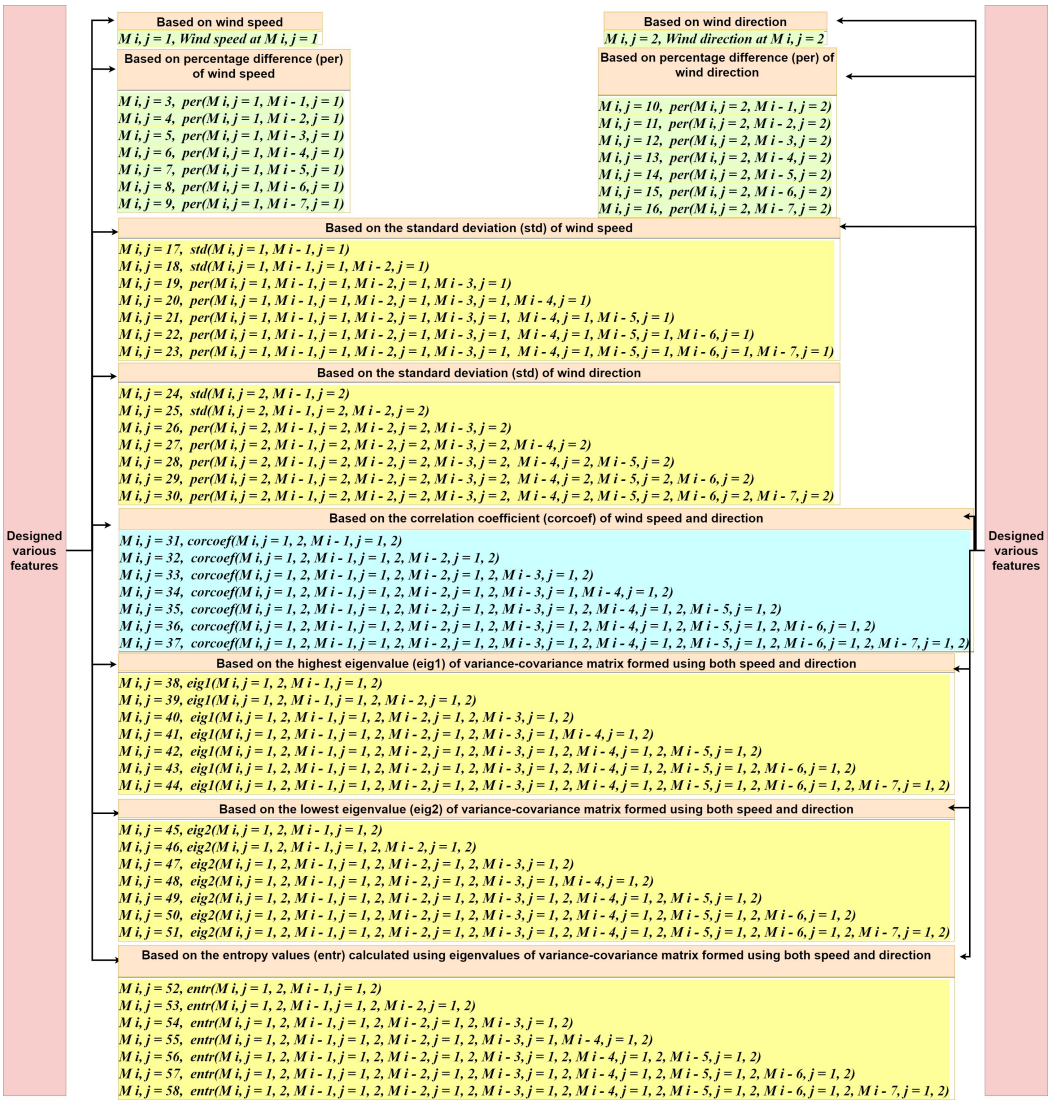


Figure 3.21: The designed various features in MCLT.

concepts provide statistical segregation of classes (Ghilani, 2010). k_i, where $i \rightarrow 1$ to 10 as shown in Table 3.3, is decided empirically. Speed values from $M_{i+K_B+1,\ j=1}$ to $M_{i+K_B+K_F,\ j=1}$ are grouped into these 21 classes, and count of values in each class is found. The class having maximum count is assigned to the output of the sample. This maximum count represents the dominant speed. Likewise, the output class of the sample based on the direction is determined by finding the maximum count of direction values from $M_{i+K_B+1,\ j=2}$ to $M_{i+K_B+K_F,\ j=2}$ among these 21 classes. Further, more training samples are designed by incrementing i from 1 to $r - K_F$, at a step of 1.

Table 3.3: The designed various classes formed using the mean and standard deviation of the wind data.

Class	Lower limit	Upper limit
1	$\mu - k_1\sigma$	$\mu + k_1\sigma$
2	$\mu + k_1\sigma$	$\mu + k_2\sigma$
3	$\mu + k_2\sigma$	$\mu + k_3\sigma$
4	$\mu + k_3\sigma$	$\mu + k_4\sigma$
5	$\mu + k_4\sigma$	$\mu + k_5\sigma$
6	$\mu + k_5\sigma$	$\mu + k_6\sigma$
7	$\mu + k_6\sigma$	$\mu + k_7\sigma$
8	$\mu + k_7\sigma$	$\mu + k_8\sigma$
9	$\mu + k_8\sigma$	$\mu + k_9\sigma$
10	$\mu + k_9\sigma$	$\mu + k_{10}\sigma$
11	$\mu + k_{10}\sigma$	$+\infty$
12	$\mu - k_2\sigma$	$\mu - k_1\sigma$
13	$\mu - k_3\sigma$	$\mu - k_2\sigma$
14	$\mu - k_4\sigma$	$\mu - k_3\sigma$
15	$\mu - k_5\sigma$	$\mu - k_4\sigma$
16	$\mu - k_6\sigma$	$\mu - k_5\sigma$
17	$\mu - k_7\sigma$	$\mu - k_6\sigma$
18	$\mu - k_8\sigma$	$\mu - k_7\sigma$
19	$\mu - k_9\sigma$	$\mu - k_8\sigma$
20	$\mu - k_{10}\sigma$	$\mu - k_9\sigma$
21	$-\infty$	$\mu - k_{10}\sigma$

MCLT Architecture

MCLT architecture is shown in Figure 3.22. There are five input layers corresponding to each view CNN_i (CNN_1, CNN_2, CNN_3, CNN_4, and CNN_5) as in the 1DM. The input layer of each view is followed by four successive convolutional layers (C_1, C_2, C_3, C_4). The densely connected convolutional layers similar to ResNet are realised as follows,

1. C_3 directly takes as input, features from both C_2 and C_1 (while in the 1DM model, C_3 took input only from previous layer C_2), and

2. C_4 directly takes input features from C_3, C_2 and C_1 (while in traditional CNN models, C_4 takes input only from C_3) (Zhao et al., 2019).

The detailed pseudo code of MCLT implementation is discussed in Algorithm 1. All the feature maps from the last convolutional layer C_4 of each view (total 5 views) are first flattened to 1D form (step 13 Algorithm 1) and then appended one after another (step 14 Algorithm 1). This appended

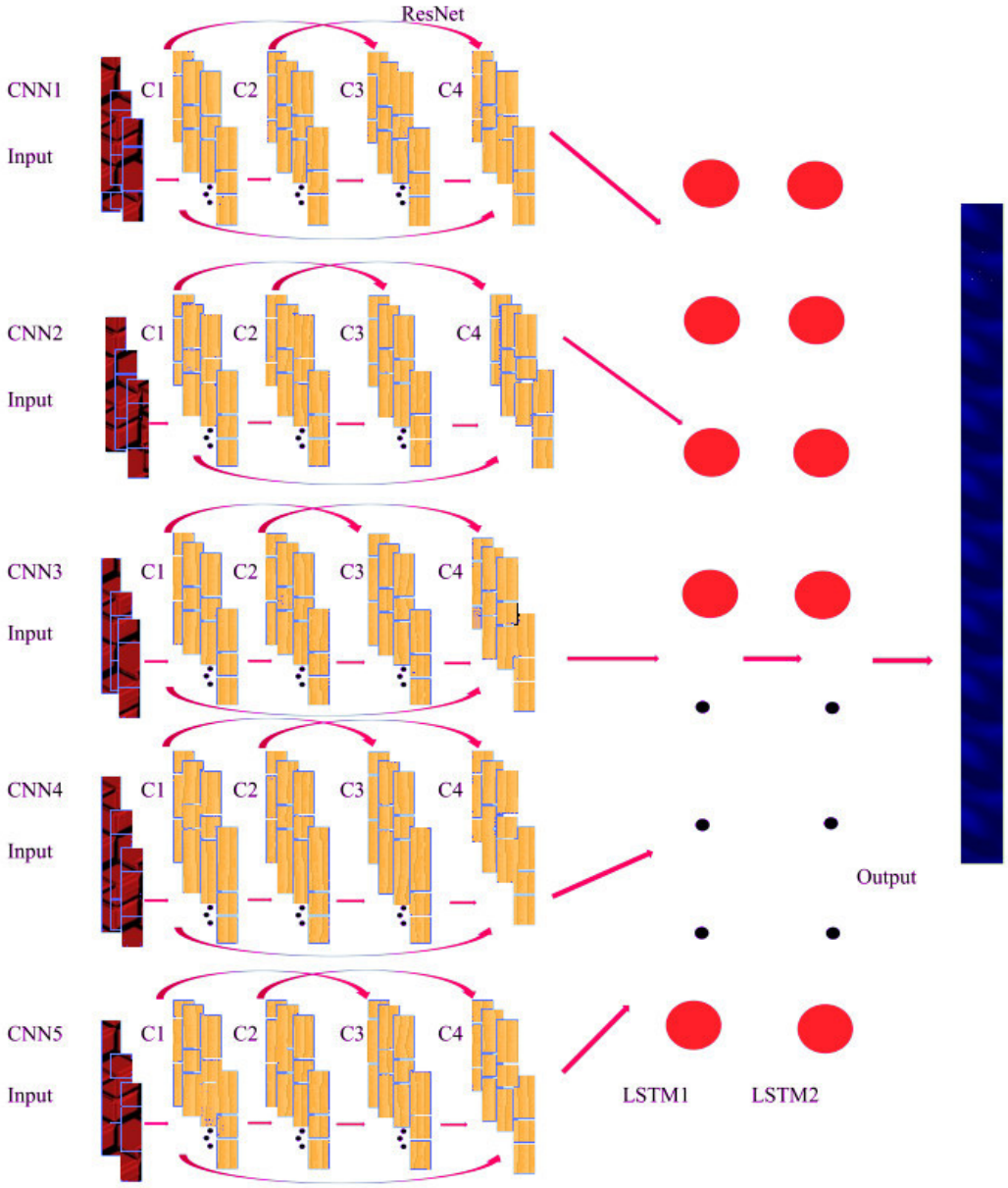


Figure 3.22: MCLT architecture. Arrows denote connections between convolutional layers and LSTM. Multiple vertical rectangles in Input, C_1, C_2, C_3 and C_4 represent multiple features in that layer.

1D feature is then passed to a common LSTM layer, called $LSTM_1$ (step 16 Algorithm 1), which in turn is followed by the second LSTM layer called $LSTM_2$. In the 1DM model, fully connected layers were present in the place of $LSTM_1$ and $LSTM_2$. The output layer comes after $LSTM_2$. The output layer uses softmax function for classification, and the number of neurons in this layer would be the same as the number of classes in the dataset (step 18 Algorithm 1).

Algorithm 1 Architecture pseudo code

1: **procedure** MCLT($Input, Output$) ▷ $Input \leftarrow$ MCLT multiple views
2: ▷ $Output \leftarrow$ MCLT output layer
3:
4: $Merged \leftarrow$ [ ] ▷ $Merged \leftarrow$ Empty list
5: **for** $i \leftarrow$ 1 to 5 **do**
6: $CNN_i processing$
7: $C_1 \leftarrow Conv1D(features, stride, input = CNN_i Input, ELU, dropout)$
8: $C_2 \leftarrow Conv1D(features, stride, input = C_1, ELU, dropout)$
9: $C_{2concat} \leftarrow Concatenate(C_1, C_2)$
10: $C_3 \leftarrow Conv1D(features, stride, input = C_{2concat}, ELU, dropout)$
11: $C_{3concat} \leftarrow Concatenate(C_1, C_2, C_3)$
12: $C_4 \leftarrow Conv1D(features, stride, input = C_{3concat}, ELU, dropout)$
13: $C_4 \leftarrow flatten(C_4)$
14: $Merged.append(C_4)$
15: **end for**
16: $LSTM_1 \leftarrow LSTM(neurons, input = Merged, dropout)$
17: $LSTM_2 \leftarrow LSTM(neurons, input = LSTM_1, dropout)$
18: $Output \leftarrow Dense(neurons, input = LSTM_2, softmax)$
19: **end procedure**

Further, $Merged$ in Algorithm 1, is initially defined as an empty list (step 4) and for each iteration inside for loop, flattened C_4 is appended to it (step 14). $CNN_i Input$ in step 7 means input corresponding to CNN_i. $Conv1D$ in Algorithm 1 denotes a function representing 1D convolutional operation, that takes values such as number of features, stride (amount by which 1D kernel shifts), input from a CNN layer, activation function and dropout (Srivastava et al., 2014) value. $Concatenate$ in Algorithm 1 means that C_1 and C_2 (step 9), C_1, C_2 and C_3 (step 11), are joined together one after another and then treated as input for the next step $i.e.$, making the densely connected convolutional layers. $LSTM$ and $Dense$ (step 16 - 18 in Algorithm 1) denote LSTM and fully connected layers, respectively. LSTM units include a memory cell that can maintain information in memory for long periods of time (Hochreiter and Schmidhuber, 1997; Karpathy et al., 2016). A set of gates is used to control when information enters LSTM units,

when it leaves, and when it is forgotten. Thus, these memory units aid in learning longer term dependencies. The densely connected convolutional layers help C_3 directly learn features from both C_1 and C_2, unlike the 1DM where C_3 learnt features from C_2 only. Likewise, C_4 directly learns features from C_1, C_2, and C_3, unlike traditional CNN where C_4 considers input only from C_3. For a given sample's input, five views corresponding to each input layer in the MCLT are formed as follows:

1. first view takes all K_B values of the sample's input $i.e.$, rows from i to $i + K_B$ (and all columns of these rows) of $M_{i,\,j}$,
2. second view takes half of K_B values of the sample's input from rows i to $i + K_B$ at an interval of two (and all columns of these rows) of $M_{i,\,j}$,
3. third view also takes half of K_B values of the sample's input but from rows i + 1 to $i + K_B$ at an interval of two (and all columns of these rows) of $M_{i,\,j}$,
4. fourth view takes one-third of K_B values of the sample's input but from rows i to $i + K_B$ at an interval of three (and all columns of these rows) of $M_{i,\,j}$, and
5. fifth view again takes one-third of K_B values of the sample's input but from rows i + 1 to $i + K_B$ at an interval of three (and all columns of these rows) of $M_{i,\,j}$.

Each input layer of the MCLT, thus, takes multiple 1D features. In the present study, there are 58 features in each input layer. A higher number of features in CNN architecture helps in learning the various properties of a sample from finer to coarser levels. Therefore many features are used in this study. Thus, for a sample having input values from i to $i + K_B$ of $M_{i,\,j}$, each column of these rows form a 1D feature of the input layer. Thus, the MCLT incorporates multiple features and multiple views in the input layers, as well as each convolutional layer takes input from several previous layers, with the presence of memory units in the LSTM layers. The output layer of the MCLT uses the sample's output class, either based on the wind speed or direction, for training and testing the architecture. The sample's output class is designed using M_{i+K_B+1} to $M_{i+K_B+K_F}$ values as discussed in the previous section.

3.3.2 Results

This section explains the results of MCLT for Stuttgart and Netherlands datasets. It section 3.3.2 provides the details of the hardware and soft-

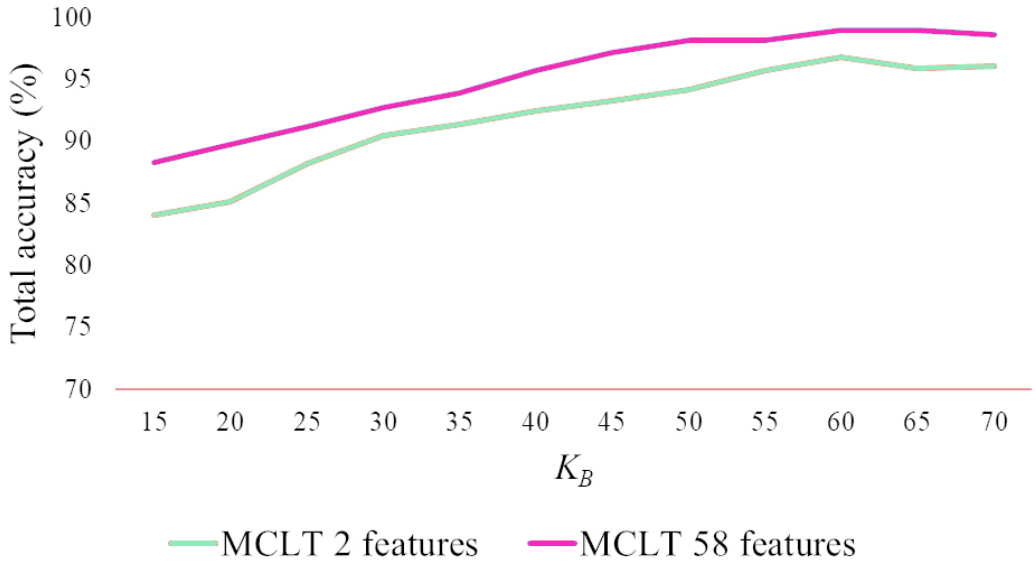


Figure 3.23: Total accuracy comparison of 2 and 58 features in MCLT for different values of K_B.

ware configuration along with the organisation of the training and testing samples. Also section 3.3.2 presents the obtained accuracies for different datasets and features. Moreover, subsection 3.3.3 represents the qualitative discussion of the obtained results and comparison with other existing methods.

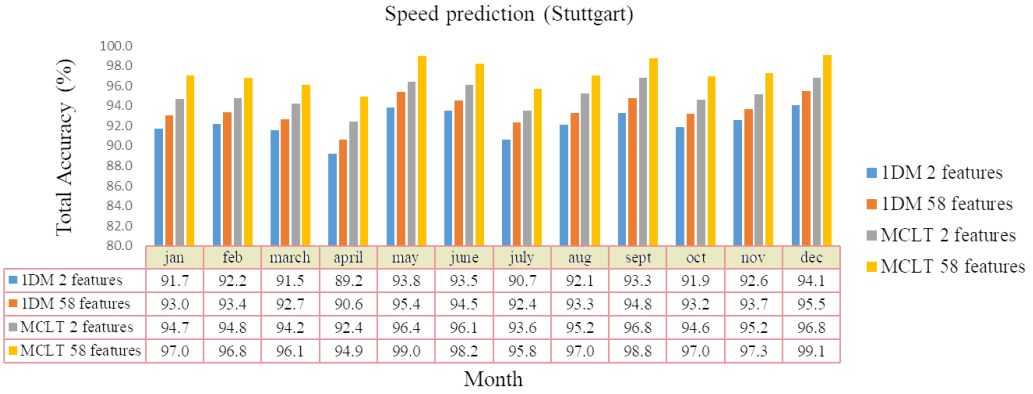
System & Samples Details

The developed MCLT architecture has been coded in Python language using Keras library (Chollet, 2017) with TensorFlow in the backend and executed on Intel® Core ™ i7- 4770 CPU @3.40 GHz having four cores. The historical temporal wind data of both the areas Stuttgart and Netherlands were separated by each month to create respective month data. In each month data, past temporal values were arranged first and then recent temporal values. $M_{i,\,j}$ is created for each month. For Stuttgart, several samples, each having input and corresponding output, were created from a month's $M_{i,\,j}$ as discussed in subsection 3.3.1. When the samples were used for predicting dominant speed, then the outputs of the samples were based on speed. Similarly, when the samples were used for predicting the dominant direction, then the outputs of the samples were based on direction. However, in both speed and direction predictions, the input of the samples remained the same. These samples created from a month's data are then used to train and test the MCLT separately for dominant speed and direction predictions. Likewise, samples were created from

each month's data of Netherlands, and the MCLT was trained and tested. The total samples for a month were randomly divided into training and testing samples, with 30% of the total samples as the testing samples. This procedure of random division of the total samples into training and testing samples, followed by the training and testing of the MCLT was repeated 20 times in order to determine the mean accuracies values. This procedure, thus, accounted for the randomness in splitting into training and testing. Further, Adaptive Synthetic Sampling (ADASYN) technique (He et al., 2008) was used to enhance the number of training samples for better learning of the MCLT. ADASYN generates samples of the minority class according to their density distributions and avoids over-sampling. The number of feature maps in C_1, C_2, C_3 and C_4 of each of CNN_1, CNN_2, CNN_3, CNN_4, and CNN_5, of the MCLT architecture are 16, 28, 32 and 32, respectively, whereas the number of neurons in $LSTM_1$ and $LSTM_2$ are 200 and 200 respectively. Values of k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9 and k_{10} (Table 3.2) were empirically determined as 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80 and 1.0 respectively (same for both speed and direction), so that sufficient number of samples occur in each class, by observing the histograms comprising 21 bins corresponding to 21 classes. Moreover, K_B and K_F were taken as 60. K_F multiplied by the temporal resolution gives a time frame of future prediction as per user desire. Figure 3.23 shows the variations in total accuracy of the MCLT with 58 features by varying K_B (here K_F = K_B). In this work, K_B is taken as 60 as accuracy increases till 60 and after that remains similar as shown in Figure 3.23. Exponential Linear Units (ELUs) (Clevert et al., 2016; Pedamonti, 2018) with α of 3.0 have been used as activation function in the MCLT. The higher value α of 3.0 was chosen to avoid dead neurons problem during training with highly variable wind datasets (Nair and Hinton, 2010; Clevert et al., 2016). Kernel size of three along with stride of one has been applied for all the convolutional layers. Batch normalisation (Jung et al., 2019) and dropout (Srivastava et al., 2014) of 0.45 have been employed after every convolution layer. This helps to prevent over-fitting, and the MCLT architecture learns better. The cross-entropy loss function has been used during training of the MCLT.

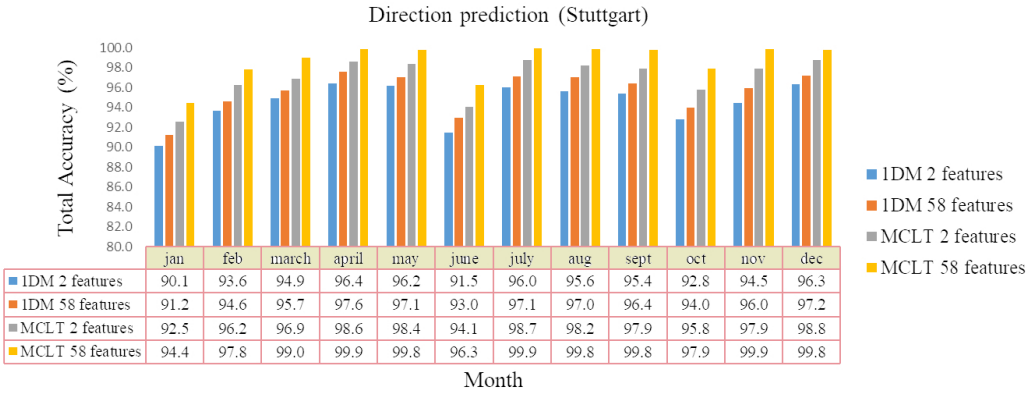
Model Accuracies

The total accuracies for different months of Stuttgart for the test samples, obtained using the MCLT are shown in Figure 3.24 and Figure 3.25. In these figures, MCLT with 58 features means that all the columns (or features) of $M_{i,\,j}$ have been used in the input layers of the MCLT, whereas MCLT with 2 features means only first two columns (of speed and direction) of $M_{i,\,j}$



	jan	feb	march	april	may	june	july	aug	sept	oct	nov	dec
1DM 2 features	91.7	92.2	91.5	89.2	93.8	93.5	90.7	92.1	93.3	91.9	92.6	94.1
1DM 58 features	93.0	93.4	92.7	90.6	95.4	94.5	92.4	93.3	94.8	93.2	93.7	95.5
MCLT 2 features	94.7	94.8	94.2	92.4	96.4	96.1	93.6	95.2	96.8	94.6	95.2	96.8
MCLT 58 features	97.0	96.8	96.1	94.9	99.0	98.2	95.8	97.0	98.8	97.0	97.3	99.1

Figure 3.24: Total accuracies in percentage for different months of Stuttgart for dominant speed prediction.

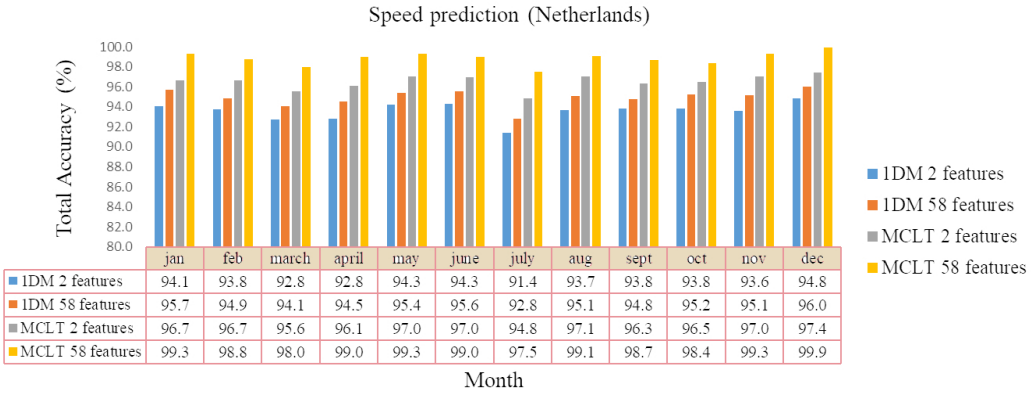


	jan	feb	march	april	may	june	july	aug	sept	oct	nov	dec
1DM 2 features	90.1	93.6	94.9	96.4	96.2	91.5	96.0	95.6	95.4	92.8	94.5	96.3
1DM 58 features	91.2	94.6	95.7	97.6	97.1	93.0	97.1	97.0	96.4	94.0	96.0	97.2
MCLT 2 features	92.5	96.2	96.9	98.6	98.4	94.1	98.7	98.2	97.9	95.8	97.9	98.8
MCLT 58 features	94.4	97.8	99.0	99.9	99.8	96.3	99.9	99.8	99.8	97.9	99.9	99.8

Figure 3.25: Total accuracies in percentage for different months of Stuttgart for dominant direction prediction.

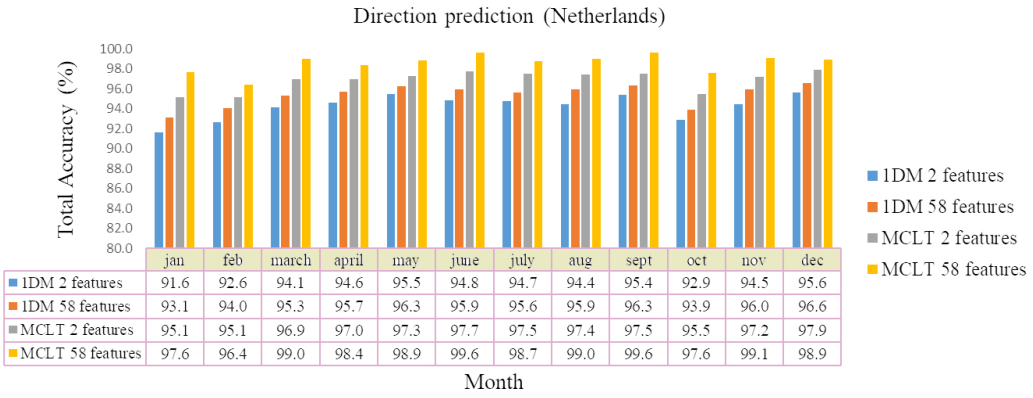
have been used in the input layers. Similar are the interpretations of 1DM with 58 features and 2 features. Figure 3.24 and Figure 3.25 represent total accuracies for dominant speed and direction prediction for different months of Stuttgart, respectively. Figure 3.26 and Figure 3.27 represent total accuracies for dominant speed and direction prediction for different months of Netherlands, respectively.

The maximum, minimum, and mean total accuracies for dominant speed prediction (for Stuttgart) using the MCLT with 58 features are 99.1%, 94.9%, and 97.2%, respectively, as shown in Table 3.4. The maximum, minimum, and mean total accuracies for dominant speed prediction (for Stuttgart) using the MCLT with 2 features are 96.8%, 92.4%, and 95.1%, respectively (Table 3.4). Similarly, the maximum, minimum, and mean total accuracies for dominant direction prediction (for Stuttgart) using MCLT



	jan	feb	march	april	may	june	july	aug	sept	oct	nov	dec
1DM 2 features	94.1	93.8	92.8	92.8	94.3	94.3	91.4	93.7	93.8	93.8	93.6	94.8
1DM 58 features	95.7	94.9	94.1	94.5	95.4	95.6	92.8	95.1	94.8	95.2	95.1	96.0
MCLT 2 features	96.7	96.7	95.6	96.1	97.0	97.0	94.8	97.1	96.3	96.5	97.0	97.4
MCLT 58 features	99.3	98.8	98.0	99.0	99.3	99.0	97.5	99.1	98.7	98.4	99.3	99.9

Figure 3.26: Total accuracies in percentage for different months of Netherlands for dominant speed prediction.



	jan	feb	march	april	may	june	july	aug	sept	oct	nov	dec
1DM 2 features	91.6	92.6	94.1	94.6	95.5	94.8	94.7	94.4	95.4	92.9	94.5	95.6
1DM 58 features	93.1	94.0	95.3	95.7	96.3	95.9	95.6	95.9	96.3	93.9	96.0	96.6
MCLT 2 features	95.1	95.1	96.9	97.0	97.3	97.7	97.5	97.4	97.5	95.5	97.2	97.9
MCLT 58 features	97.6	96.4	99.0	98.4	98.9	99.6	98.7	99.0	99.6	97.6	99.1	98.9

Figure 3.27: Total accuracies in percentage for different months of Netherlands for dominant direction prediction.

Table 3.4: The obtained maximum, minimum, and mean total accuracies for dominant speed prediction.

TA (%)	Stuttgart				Netherlands			
	1DM 2 features	1DM 58 features	MCLT 2 features	MCLT 58 features	1DM 2 features	1DM 58 features	MCLT 2 features	MCLT 58 features
Max (%)	94.1	95.5	96.8	99.1	94.8	96.0	97.4	99.9
Min (%)	89.2	90.6	92.4	94.9	91.4	92.8	94.8	97.5
Mean (%)	92.2	93.5	95.1	97.2	93.6	94.9	96.5	98.9

Table 3.5: The obtained maximum, minimum, and mean total accuracies for dominant direction prediction.

TA (%)	Stuttgart				Netherlands			
	1DM 2 features	1DM 58 features	MCLT 2 features	MCLT 58 features	1DM 2 features	1DM 58 features	MCLT 2 features	MCLT 58 features
Max (%)	96.4	97.6	98.8	99.9	95.6	96.6	97.9	99.6
Min (%)	90.1	91.2	92.5	94.4	91.6	93.1	95.1	96.4
Mean (%)	94.4	95.6	97.0	98.7	94.2	95.4	96.8	98.6

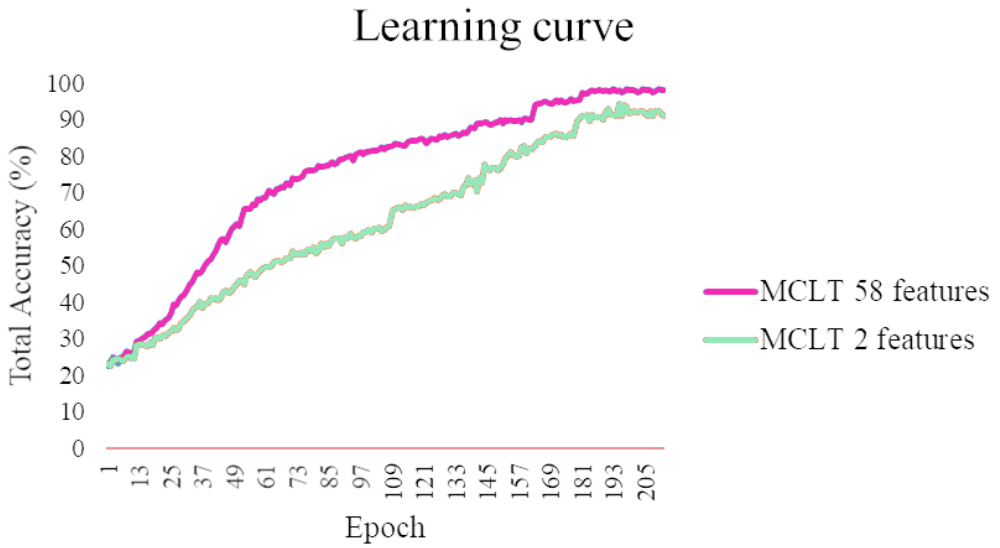


Figure 3.28: Learning curves for MCLT with 2 and 58 features.

with 58 features are 99.9%, 94.4%, and 98.7%, respectively (Table 3.5). The maximum, minimum, and mean total accuracies for dominant direction prediction (for Stuttgart) using MCLT with 2 features are 98.8%, 92.5%, and 97.0%, respectively (Table 3.5). Figure 3.24 to Figure 3.27, Table 3.4 and Table 3.5 also represent results when the 1DM architecture with 2 and 58 features is used for prediction. Learning curves and loss curves (for speed prediction) of January month's test samples of Stuttgart using the MCLT with 2 and 58 features are shown in Figure 3.28 and Figure 3.29, respectively.

3.3.3 Discussion

The developed MCLT architecture shows promising results for dominant wind speed and direction prediction of temporal wind datasets from Stut-

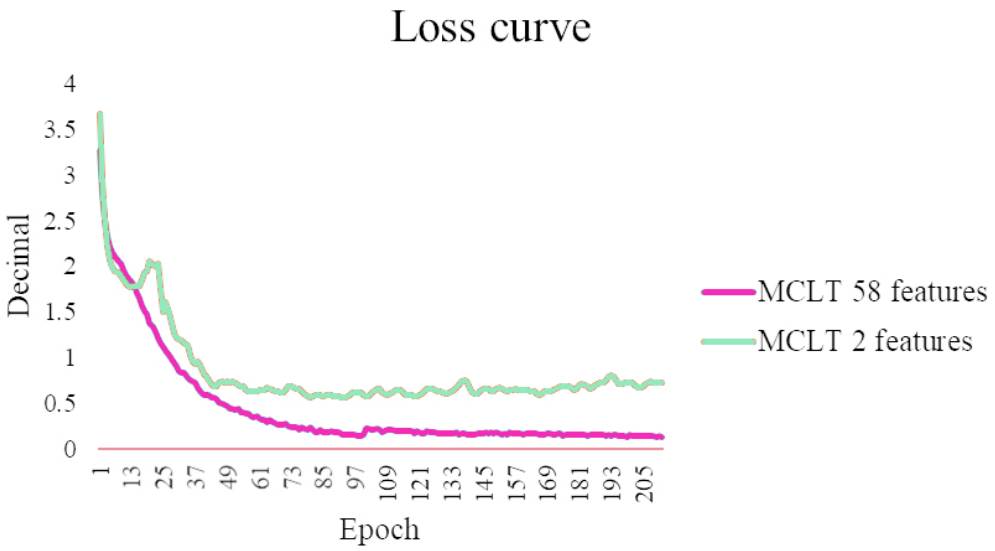


Figure 3.29: Loss curves for MCLT with 2 and 58 features.

tgart and Netherlands. Below subsections discuss the results with the help of rose plot, comparison among 2 and 58 features, and comparison with other suitable approaches.

Rose Plots

Wind rose plot helps in the visualisation of wind speed and direction in the same graph, in a circular format. The length of each spoke around the circle indicates the number of times (count) that the wind blows from the indicated direction. Colors along the spokes indicate classes of wind speed. The data of March (Mar) 2020 of Stuttgart is used to represent the real world sensor's measurements (true values) and prediction outcomes of the MCLT in Figure 3.30 and Figure 3.31, respectively. The high resemblance among Figure 3.30 and Figure 3.31, signifies that the prediction results are similar to the true (real) values. This augments visually the accuracies obtained previously in the results subsection 3.3.2. In these figures, there are 21 different colour ranges denoting the wind speed divided into 21 classses with varying spoke length and direction highlighting the wind blows count from the indicated directions in this study.

Comparison Among 2 & 58 Features

The 58 multiple features in the input layers help the MCLT learn better the temporal variations in the samples. These features are based on percentage

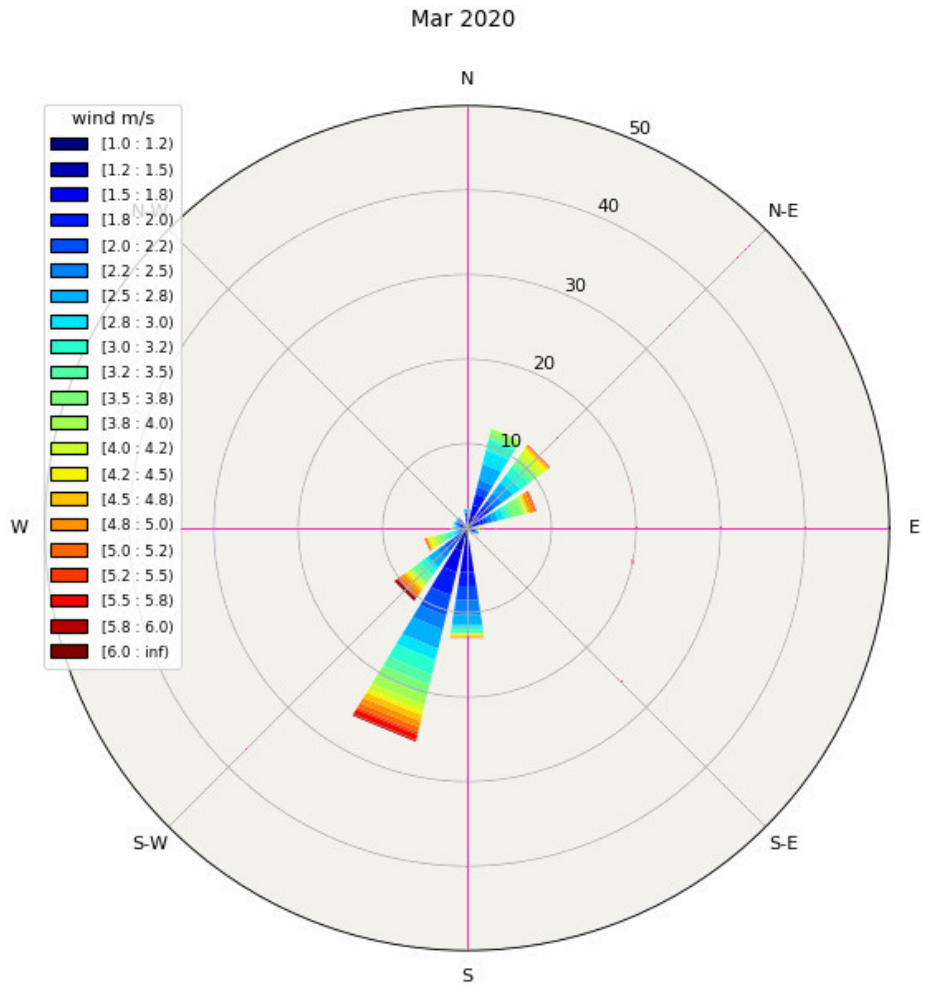


Figure 3.30: Wind rose plot for Mar 2020 (sensor's measurements).

difference, standard deviation, correlation coefficient, eigenvalues, and entropy, that are calculated by taking into account some of the nearby temporal values. As the temporal values adjacent to a time instance change, the values of these features also adapt to these changes. Thus, these features help in comprehensive description of wind speed and direction, describing the trend like increase, decrease, stationary, sudden turbulence, rate of increase and decrease, deviation from the mean, behavior of speed with respect to direction ($i.e.$, correlation), energy ($i.e.$, entropy) of the adjacent temporal values and its variation. Therefore, they provide additional information about samples. Moreover, the movements of the 1D kernels in the convolutional layers further help the convolutional layers to learn their own features in the form of weights and biases during the training phase of the MCLT. When only two features were used in the input layers of the MCLT, maximum total accuracy was 96.8% and 97.4% for Stuttgart

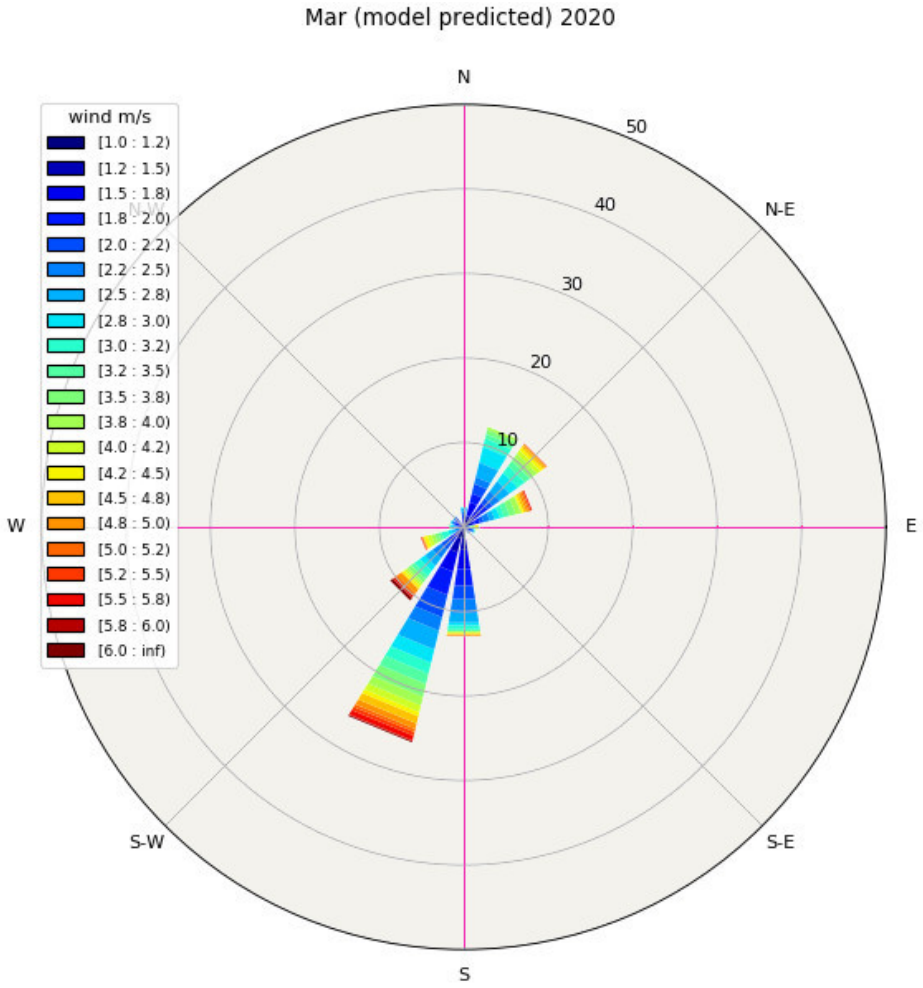


Figure 3.31: Wind rose plot for Mar 2020 (model predictions).

and Netherlands, respectively, for speed (Table 3.4) and 98.8% and 97.9% for Stuttgart and Netherlands, respectively, for direction (Table 3.5). The maximum total accuracy for MCLT with 58 features is increased by 2.3% and 2.5% for Stuttgart and Netherlands, respectively for speed (Table 3.4) and by 1.1% and 1.6% for Stuttgart and Netherlands, respectively for direction (Table 3.5) in comparison to MCLT with 2 features. Similarly, the effect of these 58 features over 2 features can also be seen in the case of 1DM (Table 3.4, Table 3.5) where maximum total accuracy for speed improved by 1.4% and 1.2% for Stuttgart and Netherlands, respectively, and by 1.2% and 1.0% for Stuttgart and Netherlands, respectively for direction. Learning of the MCLT with 58 features is better than 2 features as shown by respective learning curves in Figure 3.28 and by the loss curves in Figure 3.29.

Convolutional layers (C_1, C_2) near the input layers learn the smaller

Table 3.6: The difference in the achieved total accuracies of MCLT with 58 features and 1DM with 2 features. Positive value denotes MCLT has higher accuracy than 1DM.

TA (%)	Stuttgart		Netherlands	
	Dominant wind speed prediction	Dominant wind direction prediction	Dominant wind speed prediction	Dominant wind direction prediction
Max (%)	5.7	4.3	6.1	4.8
Min (%)	5.0	3.5	5.1	4.0
Mean (%)	5.0	4.3	5.3	4.4

features, while the convolutional layers (C_3, C_4) near the output layer learn larger features (Krizhevsky et al., 2012; He et al., 2016; Xie et al., 2017; Huang et al., 2018). C_3 takes as input the learnt features from both C_1, and C_2, while C_4, takes as input the features from C_1, C_2, and C_3, therefore, the MCLT gets trained by learning features at different scales. Further, as the convolutional layers (C_3, C_4) are connected to all the previous convolutional layers, providing that gradient vanishing problem would not occur, $i.e.$, MCLT learning does not slow down during training via back propagation (He et al., 2016; Xie et al., 2017; Huang et al., 2018). Moreover, LSTM layers after the last convolutional layers (C_4), have memory units that retain the learnt features from previous output of the neurons and operate upon them with features learnt from the current output of the neurons. This gives better learning over the fully connected layers (present in traditional CNNs) that lack these memory units. Additionally, the memory units in the LSTM help in finding correlations between patterns learnt across different time, as a recent pattern is a function of pattern learnt at previous time.

Comparison With Existing Related Work

The developed MCLT architecture is compared with the 1DM. The MCLT with 2 features as well as 58 features performs better than the 1DM with 58 features, as shown in Figure 3.24 to Figure 3.27 for both Stuttgart and Netherlands. Minimum, maximum and mean total accuracies of the MCLT with 58 features are compared with 1DM with 2 features in Table 3.6. Thus, the MCLT performs better than the 1DM. Moreover, the MCLT with 58 features efficiently predicts for the larger time frame in future (K_F as 60, multiply by the temporal wind dataset resolution) whereas the 1DM with 2 features could only predict for 50 values in future (see subsection 3.2.1).

The MCLT is also compared with the methods in the existing literature

that are near to the developed architecture. 1D CNN algorithm designed by Liu et al. (2018) has used regression technique working on the smoothed and filtered data thereby losing the originality of the wind dataset. The same samples comprising K_B = 60, input values without applying smoothening and filtering, that have been employed for the designed MCLT, are also used to train and test the regression CNN architecture (Liu et al., 2018). In this case, Symmetric Mean Absolute Percentage Error (SMAPE) (Flores, 1986) for wind speed in Stuttgart is 20.5% for K_B = 8 and reaches up to 25.5% for K_B = 60, while 14.9% for K_B = 15 and reaches up to 21.2% for K_B = 60 for wind speed in Netherlands. SMAPE of wind direction were moreover similar to these patterns. It may be noted that, here the outputs of the samples are designed using the real values ($i.e.$, regression) whereas MCLT outputs are based on the classes ($i.e.$, classification). As the future time frame of prediction increases, error also increases using the state-of-the-art CNN based regression method (Liu et al., 2018).

However, the developed MCLT based on classification shows high accuracy and mean total accuracy reaches up to 99.9% for K_B = 60, without smoothening and filtering the original wind data. Thus, the designed MCLT method gives satisfactory results for predicting dominant speed and direction for a greater time duration in the future unlike Liu et al. (2018). Limited 58 features in the input layers are only due to hardware constraints and more can be designed with more GPUs. The accuracies achieved using the designed MCLT can be further improved with better hardware resources by using a greater number of feature maps, neurons, convolutional and LSTM layers. Thus, the use of multiple features at various levels in the MCLT, viz.

1. 58 features in the input layers,
2. inputting a convolutional layer with features from all the previous convolutional layers, and
3. retaining the memory of learnt features by LSTM from previous outputs (of neurons) during training, help the designed architecture predict in future the dominating speed and direction classes with good accuracy.

Further, as the number of classes of the samples increases, detailed patterns of the nonlinear nature of the wind can be analysed but at the same time ambiguity in classification also increases. However, the designed MCLT architecture is able to overcome this ambiguity by learning multiple features and performs well even with 21 classes. The objective behind using more number of classes with close difference range helped to identify more details and results behave very close to regression with best accuracy.

3.3.4 Conclusion

In this work, a deep learning architecture is successfully designed and demonstrated to predict the dominant speed and direction classes in the future for the temporal wind datasets. The developed MCLT architecture uses 58 multiple features in the input layers, that are designed using wind speed and direction values. These features are based on percentage difference, standard deviation, correlation coefficient, eigenvalues, and entropy, for comprehensively and efficiently describing the wind trend and its variations. LSTM layers at the end of the last convolutional layers, have memory units that employ features learnt during current as well as the previous output of the neurons. Further, densely connected convolutional layers in the MCLT help the convolutional layers to learn features of other convolutional layers as well. Two large wind datasets from Stuttgart and Netherlands are used for training and testing the MCLT. The maximum total accuracies for speed and direction prediction are 99.9% and 99.9%, respectively. The average total accuracies reach up to 98.9% and 98.7%, for speed and direction prediction, respectively. The model's real world prediction demonstration support the novelty of the work while explaining visually with the help of wind rose plots. Thus, the MCLT shows promising results for different wind datasets. The limited hardware resources restricted this study in using 58 features in the input layers. However, in the above discussed ML based methods for prediction, there is a lack of visualisation as required in VA. Thus, an approach is required that helps in the visualisation of different patterns in the dataset for different time frames.

3.4 Chapter Summary

In this chapter, several deep learning architectures have been developed to provide a comprehensive framework to perform the prediction analyses of meteorological and pollution parameters. The first approach develops three (comparative) One Dimensional (1D) algorithms using Long Short Term Memory (LSTM), Random Forest (RF) and Support Vector Machine (SVM) for dominant wind speed and direction prediction. The developed 1D LSTM (1DLSTM), RF (1DRF) and SVM (1DSVM) take successive time values in terms of wind speed and direction as input and predict the future dominant speed and direction, separately. The developed algorithms are trained and tested using the historical wind dataset of Stuttgart and Netherlands. Prediction using 1DLSTM results in total accuracies reaching up to 94.0%, up to 92.0% using 1DSVM and up to 88.0% using 1DRF for speed and direction.

Previous accuracies are improved upon in the next developed algorithms based on Convolutional Neural Networks (CNNs). The concept is advanced and implemented progressively for more environmental data (meteorological and pollution parameters), and their effects integrated together in the following work. The 1D Single CNN (1DS) takes as input the consecutive temporal values in terms of the wind speed and direction and predicts in future dominating speed and direction, separately, after the last value in the input. The 1D Multiple CNN (1DM) combines several 1DS but with different views of the same input, therefore, learning more information compared to the 1DS. Total accuracies reached up to 95.2%, 95.1% for predicting the dominant wind speed and direction, respectively, using the 1DS and up to 98.8%, 99.7% for predicting the dominant wind speed and direction, respectively, using the 1DM. Unlike other (existing) methods that use regression techniques with manually designed features to predict speed and direction, the developed methods have used classification techniques with the 1DS and 1DM learning their features automatically on the original environmental dataset.

Moreover, this chapter has also successfully demonstrated an approach based on a multiple CNN architecture with multiple input features, combined with multiple LSTM, along with densely connected convolutional layers ($i.e.$, MCLT), for temporal wind nature analysis. Multiple features (total 58) features in the input layers of the MCLT, are designed using wind speed and direction values. These empirical features are based on percentage difference, standard deviation, correlation coefficient, eigenvalues, and entropy, for efficiently describing the wind trend. Two successive LSTM layers are used after four densely connected convolutional layers of the MCLT. Moreover, LSTM has memory units that utilise learnt features from the current as well as previous outputs of the neurons, thereby enhancing the learning of patterns in the temporal wind dataset. The presence of a densely connected convolutional layer helps to learn features of other convolutional layers as well. The maximum and minimum total accuracies for dominant speed prediction are 99.1% and 94.9%, (for Stuttgart) and 99.9% and 97.5% (for Netherlands) and for dominant direction prediction are 99.9% and 94.4% (for Stuttgart) and 99.6% and 96.4% (for Netherlands), respectively using MCLT with 58 features. The wind rose plot analyses are also performed to deliver clarity of the designed model. The MCLT therefore, with multiple features at different levels, $i.e.$, the input layers, the convolutional layers, and LSTM layers, shows promising results for the prediction of dominant speed and direction as a classification method. The developed framework is implemented for the Stuttgart and Netherlands sensors locations. However, it can be applied to any number of sensors for any given location (area) with some ML tuning and training of the respect-

ive datasets. The results of the MCLT, 1DS and 1DM are analysed based on a lesser (2) and higher (58) number of features in the designed input layers, as well as accuracy variation with different sizes of the forward and backward windows has been plotted for these algorithms. Also, learning curves have been plotted. These aspects contribute to XAI domain (Choo and Liu, 2018).

4

Seasonality Deduction Application

This chapter presents seasonality deduction approach designed in this work. The following section 4.1 explains the setup of developed seasonality deduction platform. The core components of results and findings are given in subsection 4.1.3 along with the elaborate discussion of the results in subsection 4.1.4. Furthermore, the chapter summary (section 4.2) at the end gives an overview of the concepts of this chapter in a simplified way.

Developed Methods The approach considers the hourly time series Particular Matter (PM) $PM_{2.5}$ and PM_{10}, Nitrogen Oxide (NO), and Nitrogen Dioxide (NO_2), and Ozone (O_3) along with the measured wind flow and humidity. The study's objective is to assess the temporal seasonality patterns of these parameters in Stuttgart, Germany. The temporal variations over the city center in Stuttgart are analysed using unsupervised approach to perform seasonal hierarchical clustering on a series of parameters NO, NO_2, O_3, PM_{10}, and $PM_{2.5}$, wind speed and humidity. Furthermore, the correlations between meteorological and pollution parameters are analysed using the Spearman rank correlation method. Moreover, a dashboard is

Parts of this chapter have previously been published in:

Harbola, S. and Coors, V. (2018), 'Geo-Visualisation and Visual Analytics for Smart Cities: A Survey', *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-4/W11, https://doi.org/10.5194/isprs-archives-XLII-4-W11-11-2018, 11–18;

Harbola, S. and Coors, V. (2020), 'Seasonality Deduction Platform : For PM_{10}, $PM_{2.5}$, NO, NO_2 and O_3 in Relationship with Wind Speed and Humidity', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, VI-4/W2, https://doi.org/10.5194/isprs-annals-VI-4-W2-2020-71-2020, 71–78;

Harbola, S., Storz, M., and Coors, V. (2021b), *Augment Reality for Windy-cities:3D Visualisation of future wind nature analysis in city planning* (Springer, (To appear, accepted on 2020- July -20)).

developed to provide the user with the desired time frame of visualisation of these parameters.

4.1 Overview: Seasonality Deduction Platform

The human activities not only contributed to the lifestyle advancement and developments, meanwhile also to pollution, and change in the climate as byproducts. Very small aerial pollutants are discharged from the chimneys, industrial waste, vehicle smokes, and construction sites, that can be inhaled with the air leading to heart diseases, lung and respiratory problems all over the world. The traffic related pollutants like Particular Matter (PM) $PM_{2.5}$ and PM_{10}, Nitrogen Oxide (NO) and Nitrogen Dioxide (NO_2), and Ozone (O_3) remain at a high level. The air quality is affected by, NO, NO_2, O_3, PM_{10}, $PM_{2.5}$ and their atmospheric concentrations. The lung tissue damage, cardiovascular and chronic respiratory diseases, could be hassled by coming in exposure to PM_{10} and $PM_{2.5}$ $i.e.$, particles with aerodynamic diameters less than 10 and 2.5 μm, respectively (Chen and Zhao, 2011). Over the urban areas, the elevated levels of pollution parameters are incorporated with both local emission sources and regional transportation (Chen and Zhao, 2011; Jasen et al., 2013). Regional transportation with diesel vehicles are the main sources of particular matters and contribute a significant portion to their levels (Wallace and Hobbs, 1977). Many studies have been performed to discover the seasonality of the pollution parameters along with the meteorological datasets, $e.g.$, wind speed, wind direction, temperature, humidity, precipitation, pressure.

Some existing literature concluded that when the wind speeds were lower than 3.5 m/s, and the temperature was higher than 21.1 °C than often high concentrations of PM_{10}, and $PM_{2.5}$ were detected with reference to a study of PM in Ohio USA (Arthur and Owen, 2003; Fraser et al., 2003). Moreover, some researchers emphasised that pollution parameters are correlated to humidity and wind flow during winter (Elminir, 2005). Hien et al. (2002) showed that wind speed, and temperature highly control the concentration of particulate matter. Few studies link pollutant characteristics to the meteorological parameters as with wind effects and humidity (Garrett and Casimiro, 2011). Several above discussed studies used smoothening and filtering techniques, ignoring the data noise and modifying the originality of temporal dataset. The comprehensive study of meteorological parameters and their contribution to $PM_{10\text{-}2.5}$, NO, NO_2, O_3 are poorly understood. Above research suggests that there is still a number of questions that remain to be addressed such as temporal wind nature and pollution parameters correlations, how humidity governs the $PM_{10\text{-}2.5}$, and

NO, NO_2, O_3 relationships for user desired time frame, without modifying the authenticity of the original temporal dataset.

A better insight into the system is required by improving human interaction with the meteorological data (see section 2.3) in relationship with pollution parameters. Thus, this motivates the current research. The problem of air pollution has caused considerable public concern in Stuttgart (Germany). Therefore, investigations into the spatio-temporal variation of concentrations of $PM_{10\text{-}2.5}$ and gaseous pollutants across Stuttgart are necessary and essential. To keep track of the mass concentrations of $PM_{10\text{-}2.5}$, NO, NO_2, O_3, these parameters have been monitored in all important cities of Germany. Data from provincial and more effective center weather monitoring in Stuttgart were selected. Temporal variations of meteorological and pollution parameters were assessed and their trends of variation between each other with respect to time for Stuttgart were investigated. Thus, unsupervised hierarchical clustering and correlation method which work on the original temporal datasets by taking into consideration the above listed gaps, are still required. Therefore, the current study proposes hierarchical clustering and Spearman rank correlation method with the following contributions:

1. in depth temporal analysis of pollution and meteorological parameters using hierarchical clustering method, without applying any smoothing and noise removal technique on the collected temporal dataset,
2. the time frame of analysis is user-defined,
3. dendrogram and heatmap temporal dataset visualisation to highlight the behavior of these parameters and to enhance accuracy, and
4. comparative study of the pollution parameters and their effects with interactive dashboard view.

The developed work would provide foreknowledge of meteorological parameters nature in relationship to pollution parameters of an area, thereby helping and supporting in optimal selection of green sites with highlighting and tuning the air pollution quality. This would encourage more utilisation of renewable energy for safe and better city planning, which in turn would help for efficient management and development of the city's green resources. The increasing air pollution in big industrial cities would be alarmed and reduced for the future with this analysis. The remaining sections are organised as follows, designed methods and datasets employed are discussed in subsection 4.1.1 and subsection 4.1.2,

respectively, subsection 4.1.3 and subsection 4.1.4 demonstrate the results and discussion, followed by conclusion in section subsection 4.1.5.

4.1.1 Methodology

The developed method analysed seasonality in seven parameters using hierarchical clustering and Spearman rank correlation. Initially, the values of each parameter are preprocessed before applying the clustering. The preprocessing involves normalising of the data followed by temporal filtering. The mean and standard deviation of a parameter are calculated. The values of a parameter are then subtracted by mean, followed by division with standard deviation, to get the normalised value. Further, the temporal filtering is applied on these normalised values. In the current study, the temporal filtering based on four quarters in a year is applied. First-quarter Q_1 is spring (March to May), second-quarter Q_2 is summer (June to August), third-quarter Q_3 is autumn (September to November), and fourth-quarter Q_4 is winter (December to February). These four time quarters divisions help in depth seasonality analysis of the considered seven parameters. Unsupervised agglomerative hierarchical clustering is applied on the temporal dataset (values) of a quarter ($i.e.$, the output of temporal filtering). The proximity matrix in hierarchical clustering helps in identifying the similarity of the clusters and combines most similar clusters hierarchically until the desired number of clusters are obtained. Ward's method in hierarchical clustering minimises the variance within the cluster by using the objective function of the error sum of squares (Ward, 1963). The pair of clusters that leads to a minimum increase in total within cluster variance after merging is searched. This increase is a weighted squared distance (D) between cluster centers (A_i, A_j) as shown in equation. 4.1 (Cormack, 1971).

In order to provide more detailed comparison and seasonality trends analysis, each quarter is considered for all the parameters. The quarter has been divided into two sets of 15 days starting and 15 days back. The sum of the squares starting from the clusters found by Ward's method is kept minimised. This gives a hint through the merging cost. The number of clusters keeps on reducing until the merging cost increases and then used the cluster number, right before the merging cost increased simultaneously (Paul and Murphy, 2009). Moreover, a dendrogram is used to obtain the final number of clusters as k. The dendrogram is a technique of agglomerative hierarchical clustering that gives a tree like diagram that records the sequences of merges or splits. In addition Spearman rank correlation analysis between the meteorological and pollution parameters helps to derive the relationship among these parameters. Spearman rank correlation

is defined in equation. 4.2, where d^2 represents square of the difference, ρ is the correlation coefficient, n is the number of measurements, and k is the number of clusters.

$$D_{\mathrm{i,j}} = D(A_{\mathrm{i}}, A_{\mathrm{j}}) = ||A_{\mathrm{i}} - A_{\mathrm{j}}||^2 \tag{4.1}$$

$$\rho = 1 - \frac{6 \sum {d^2}_i}{n(n^2 - 1)} \tag{4.2}$$

Moreover, an interactive dashboard is developed to provide in depth analytic and seasonality patterns clarity in between the meteorological and pollution parameters for user desired inputs in the four time quarters. This dashboard is called as seasonality analysis kit. The user could select the parameters over the desired time frame and compare the patterns interactively. The interactive dashboard is still in the first phase and would be more refined in future work. The developed work provides a comprehensive understanding of the relationship among the pollution parameters like NO, NO_2, O_3, PM_{10}, $PM_{2.5}$, and the meteorological parameters such as wind flow and humidity.

4.1.2 Dataset Used

Stuttgart pollution parameters and meteorological temporal datasets are used in this study. In the corner of Hauptstaetter Strasse 70173 Stuttgart, the historical data from 2015 to 2019 are taken from central Stuttgart station sensor (Stadtklima-Stuttgart, 2021). This dataset contains the wind (speed and direction) and humidity along with NO, NO_2, O_3, PM_{10}, $PM_{2.5}$, with temporal information attached in a 30 minute time interval. Amongst multiple values of a parameter in a single day, the mean value is considered in this study. The area's dataset is organised separately into an individual month by using time information, with past data first, followed by current data then subdivision into four considered quarters Q_1, Q_2, Q_3, and Q_4. This helps to perform pollution parameters and meteorological temporal datasets seasonality test and in depth analysis.

4.1.3 Results: Use Case

The developed seasonality analysis was implemented using Python and executed with four cores on Intel® Core ™ i7- 4770 CPU @3.40 GHz. Stuttgart's 2015 to 2019 years of historical data with a temporal resolution of 30 minute was separated by month to create monthly data over the years for both meteorological and pollution parameters. Figure 4.1 and Figure 4.2, show the data values recorded in a day over the 2015 to 2019

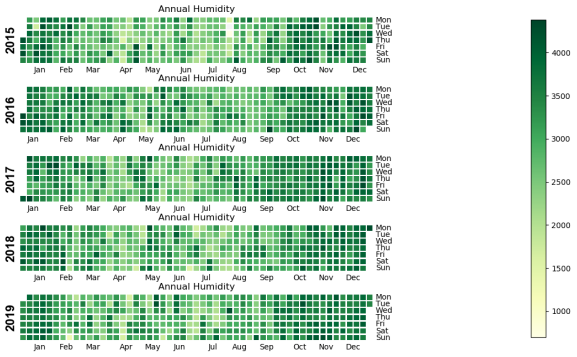


Figure 4.1: Annual humidity data value per day over the years (2015 to 2019).

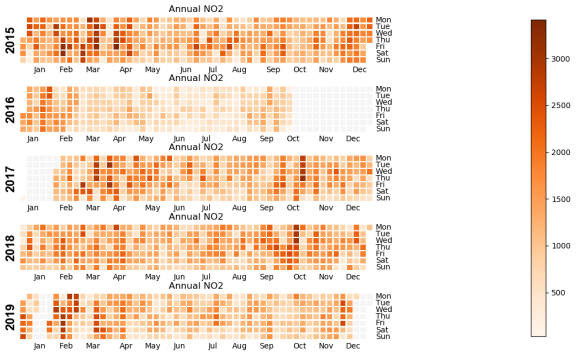


Figure 4.2: Annual NO_2 data value per day over the years (2015 to 2019).

years in the heat maps representations for humidity and NO_2 respectively. In these generated heat maps, the intensity of the color was governed by the magnitude of parameter values. A similar heat map display existed for other parameters as well. The selected parameter (anyone $i.e.$, wind speed, direction and humidity along with NO, NO_2, O_3, PM_{10}, $PM_{2.5}$), having higher values (range) over the time, had been assigned a darker color in the respective heat map. An unsupervised approach was used to perform comprehensive seasonal hierarchical clustering on a series of meteorological and pollution parameters.

The comprehensive analysis for seasonality was studied based on four quarters (Q_1, Q_2, Q_3, Q_4) over the years. In performing the hierarchical clustering, k was taken as 6. This value of k was found empirically by performing some sensitivity tests, like,

1. if the value of k was higher ($i.e.$, number of clusters was equal to the total values in a quarter) than the clustering outcome was

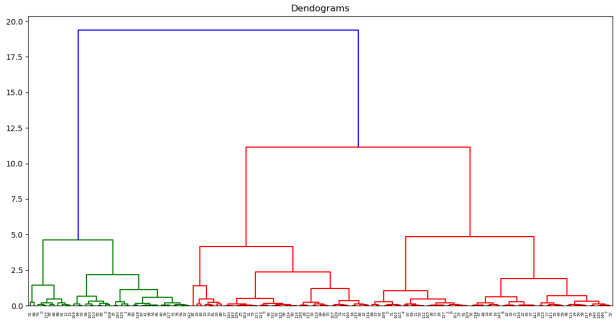


Figure 4.3: Dendrograms for selecting clusters in the temporal data set (here humidity as a considered parameter).

similar to Figure 4.1 and Figure 4.2, and this was not able to represent the seasonality pattern,

2. if the value of k was lower ($i.e.$, k = 1, 2, 3, 4), then also there was information loss, and

3. the dendrograms were generated as an output from unsupervised hierarchical clustering with the primary use to allocate objects to clusters in the best possible way.

Figure 4.3 shows the obtained dendrogram for selecting clusters (possible numbers) in the temporal data set, where in this Figure 4.3, $e.g.$, the humidity was considered. Similar parameter analyses were conducted for rest of the parameters. Therefore k was taken as 6 in the present study. The unsupervised hierarchical clustering here aimed at inferring the inner structure and trends present within the meteorological and pollution data, trying to cluster them into six classes depending on similarities among them.

In order to provide a more detailed comparison and seasonality trends analyses, quarter time frames were considered for all the parameters. Further, a quarter was divided into two parts comprising of the first fifteen days and the last fifteen days in a quarter. This helped in discovering all the possible changes in the quarter for each of the considered parameter. The obtained outputs of the in depth unsupervised clustering analysis performed for NO_2, are represented in Figure 4.4, and Figure 4.5 where the clustering outputs for NO_2 in first and last 15 days for Q_1 are shown and, Figure 4.6, and Figure 4.7 show clustering outputs for Q_2. Similarly, Figure 4.8, and Figure 4.9 depict the clustering outputs for NO_2 for first

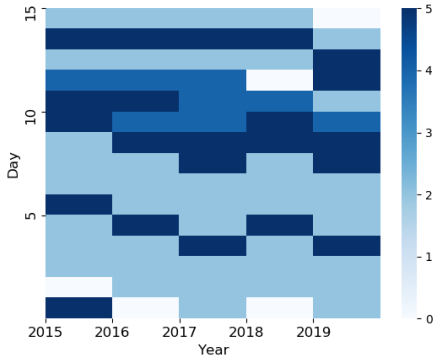


Figure 4.4: Clustering output for NO_2 for first 15 days in Q_1 over 2015 to 2019.

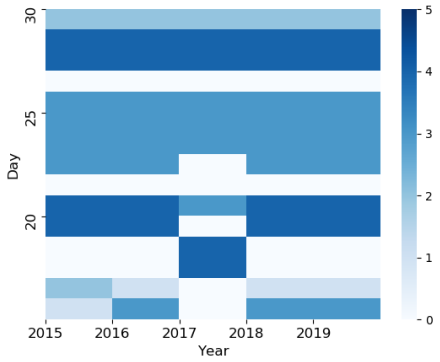


Figure 4.5: Clustering output for NO_2 for last 15 days in Q_1 over 2015 to 2019.

and last 15 days in Q_3, and Figure 4.10, and Figure 4.11 show clustering outputs for Q_4. Like these hierarchical clustering outputs, similar outputs were generated for other parameters in each respective quarters with the first and last fifteen days comparisons.

Further, the correlation analysis between the meteorological and pollution parameters were done to enhance the probability of deriving the relationships among these parameters. Figure 4.12 helps to study the complex relationships among parameters very well. In addition, the user could select the parameters over the desired time frame and compare the patterns interactively with the help of the developed dashboard. The screenshots of the designed dashboard are shown in Figure 4.13, where wind speed ($e.g.$, case) was selected as a parameter with respect to Q_1, Q_2,

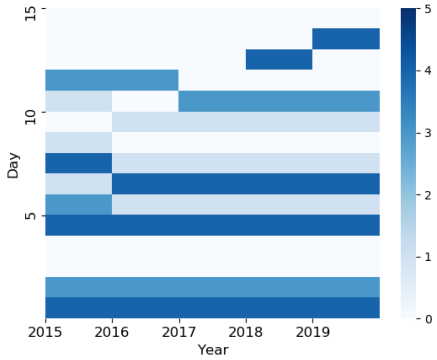


Figure 4.6: Clustering output for NO_2 for first 15 days in Q_2 over 2015 to 2019.

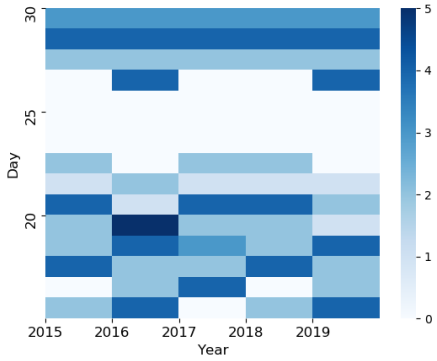


Figure 4.7: Clustering output for NO_2 for last 15 days in Q_2 over 2015 to 2019.

Q_3, Q_4 over the years to visualise seasonality. Similarly more parameters could be selected from the seasonality analysis kit.

4.1.4 Discussion

The hierarchical cluster analyses for meteorological and pollution parameters were done to highlight the trends at which any given pair of quarters (over the years) joined together in clustering diagram with each class assigned a specific color code. A sequential scale of color brewer blues scale color map was used for showing classes (0 to 5) with the color frequency differentiating low values class from high values class. The blended progression using typically of a single hue, from the least to the most opaque

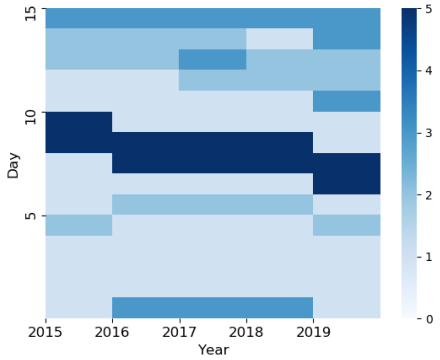


Figure 4.8: Clustering output for NO_2 for first 15 days in Q_3 over 2015 to 2019.

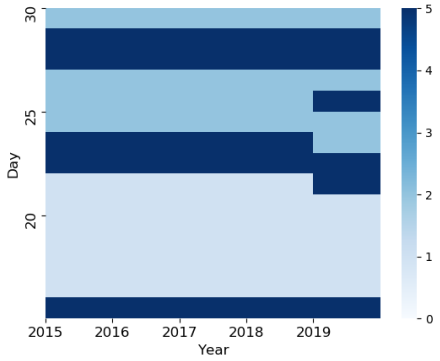


Figure 4.9: Clustering output for NO_2 for last 15 days in Q_3 over 2015 to 2019.

shades, represents low to high values. Each year dataset for the considered parameter over the four quarters that joined together sooner (in clustering) are more similar to each other than those that are joined together later. The total within cluster variance is minimised during clustering. At each step, the paired clusters with minimum between cluster distance are merged. As a result it is observed that NO and NO_2 concentrations are high in Q_3 autumn, and Q_4 winter over 2015 to 2019 respectively (Figure 4.8, and Figure 4.9, Figure 4.10, and Figure 4.11). Both are strongly correlated to each other with similar trends over the years, also same can be seen in the correlation graph in Figure 4.12. The comparison of Figure 4.4 and Figure 4.5 provides that in Q_1 there exits volatility in the first and last fifteen days. From 3^{rd} to 6^{th} day during 2015 to 2019 there exist a pattern

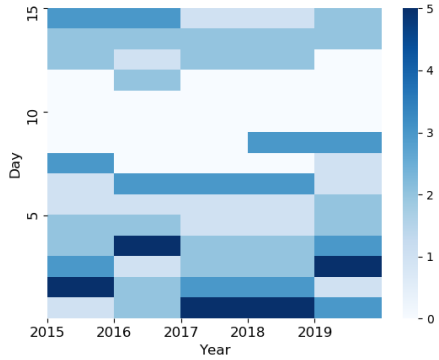


Figure 4.10: Clustering output for NO_2 for first 15 days in Q_4 over 2015 to 2019.

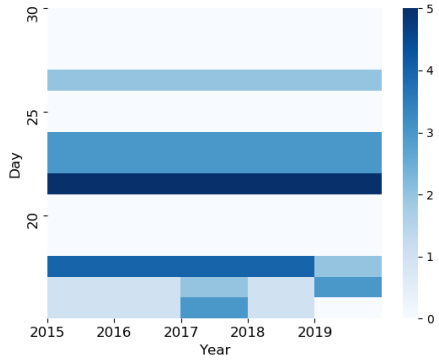


Figure 4.11: Clustering output for NO_2 for last 15 days in Q_4 over 2015 to 2019.

with high NO_2 concentrations. The same pattern repeats again from 8^{th} to 9^{th} in 2016 to 2019 and on 14^{th} in 2015 to 2018. However, in the last fifteen days from 21^{st} to 22^{nd}, and 26^{th} to 27^{th} for 2015 to 2019, low magnitudes of NO_2 are measured for Stuttgart. As shown in Figure 4.6, and Figure 4.7 from 12^{th} to 15^{th} in Q_2 NO_2 concentrations are lowest during 2015 to 2017 and, reached highest in 2018 to 2019. For last fifteen days from 16^{th} to 19^{th} the concentrations reached highest during 2016 to 2019. However, from 25^{th} to 27^{th} NO_2 measurement was negligible in 2015 to 2018, with exceptional high concentrations during 2016 and 2019. The first and last days clustering output (in Figure 4.8, and Figure 4.9) for Q_3 from 8^{th} to 10^{th} recorded high values again in 2017 to 2019. On the other hand from 17^{th} to 23^{rd} NO_2 concentrations are low for 2015 to 2018 but, measured

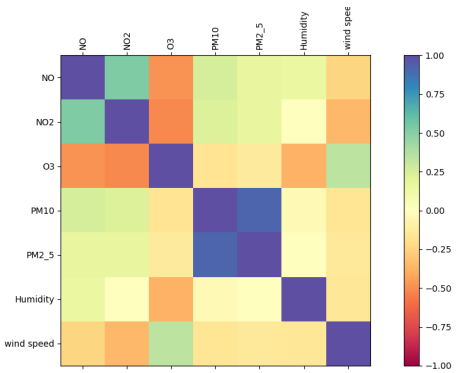


Figure 4.12: Correlation output between meteorological and pollution parameters.

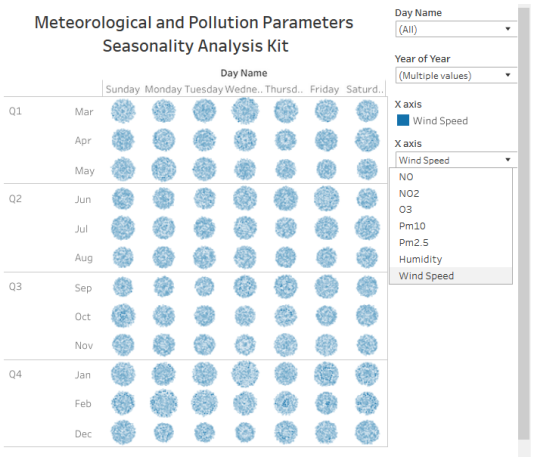


Figure 4.13: Interactive dashboard for meteorological and pollution parameters.

highest in 2019. Figure 4.10, and Figure 4.11 conclude that in Q_4 from 8^{th} to 10^{th} in 2015 to 2017 the concentrations are lowest and high in 2018 to 2019. Moreover, from 21^{st} to 22^{nd} in 2015 to 2019 the NO_2 concentrations approached highest again.

However, O_3 concentrations are more in Q_1 spring, Q_2 summer, and less in Q_3 autumn with exceptional increase in Q_4 winter during 2015 to 2019. These (above) statements also validate that O_3 and NO_2 are negatively correlated to each other which also supports the obtained correlation in Figure 4.12. Further, O_3 concentration analysis for Q_1 has been shown in Figure 4.14, and Figure 4.15, where from 1^{st} to 3^{rd} day concentrations are highest in 2016, 2018 and 2019. O_3 concentrations approach lowest from

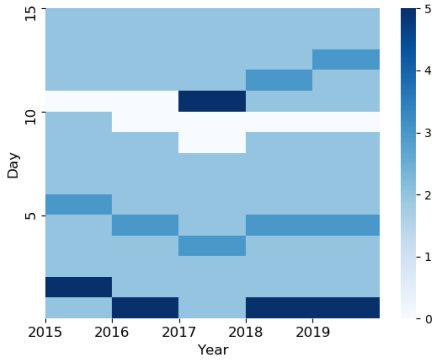


Figure 4.14: Clustering output for O_3 for first 15 days in Q_1 over 2015 to 2019.

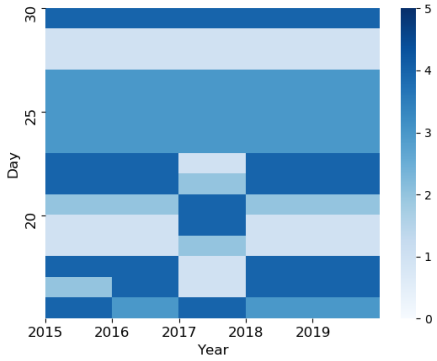


Figure 4.15: Clustering output for O_3 for last 15 days in Q_1 over 2015 to 2019.

27^{th} to 29^{th} in 2015 to 2019. As shown in Figure 4.16, and Figure 4.17, in Q_2 from 13^{th} to 14^{th} and 25^{th} to 27^{th} day O_3 concentrations are highest in 2015 to 2019. During Q_3 with reference to Figure 4.18, and Figure 4.19, from 8^{th} to 10^{th} O_3 concentrations are lowest during 2015 to 2018 while measuring highest in 2019. From 26^{th} to 27^{th} the concentrations are increasing during 2015 to 2019. O_3 variation in Q_4 is shown in Figure 4.20, and Figure 4.21, where O_3 is high from 8^{th} to 13^{th}, 16^{th} to 21^{st} and 25^{th} to 30^{th} and reaches highest in 2019. Moreover, humidity magnitudes are lowest in Aug and then starts increasing from Sep $i.e.$, in Q_3 autumn to Q_4 winter, over the years 2015 to 2019 as shown in Figure 4.26, and Figure 4.27. This shows that humidity is negatively correlated to O_3, however, positively correlated to NO_2 that also get justified by the correlation graph in Figure 4.12. Humidity

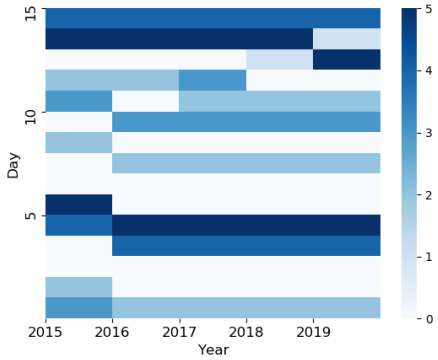


Figure 4.16: Clustering output for O_3 for first 15 days in Q_2 over 2015 to 2019.

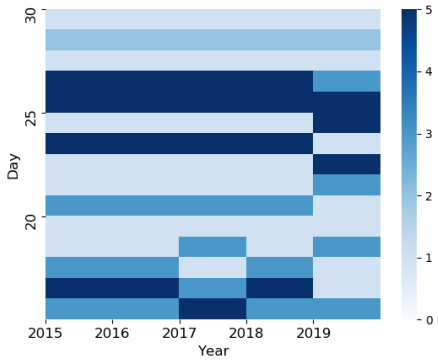


Figure 4.17: Clustering output for O_3 for last 15 days in Q_2 over 2015 to 2019.

clustering output delivered that in Q_1 (in Figure 4.22, and Figure 4.23) from 10^{th} to 12^{th}, and 26^{th} to 27^{th} the humidity measurements are highest in 2015 to 2019. As shown in Figure 4.24, and Figure 4.25 for Q_2 from 3^{rd} to 4^{th}, 9^{th} to 11^{th}, and 28^{th} to 29^{th} highest humidity was measured over 2015 to 2019. Moreover, from 10^{th} to 15^{th} humidity measured lowest in 2015 with sudden increasing spikes in 2016 to 2019 for Q_3 (in Figure 4.26, and Figure 4.27). Similarly, from 25^{th} to 28^{th} the humidity increased to highest during 2015 to 2019. With reference to Figures[1], in Q_4 from 10^{th} to 14^{th}, and 19^{th} to 22^{nd}, measured humidity increased in 2015 to 2019.

Moreover, wind datasets are highly volatile in nature over the years 2015 to 2019. High magnitude wind speeds are recorded more during Q_2

[1]Figures available: GitHub `https://www.github.com/shharbola/SDSC20_Images/`

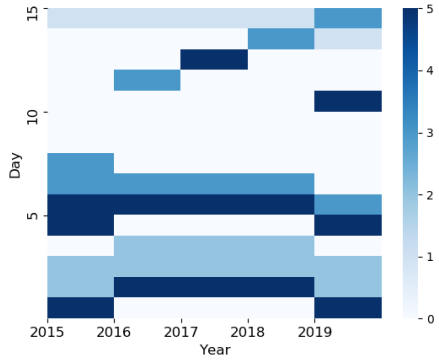


Figure 4.18: Clustering output for O_3 for first 15 days in Q_3 over 2015 to 2019.

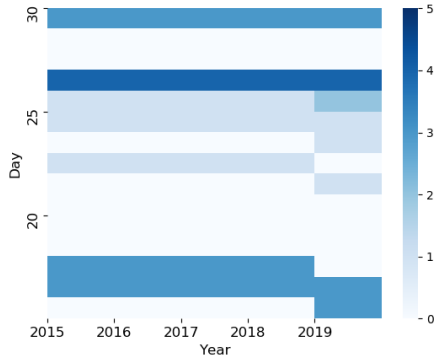


Figure 4.19: Clustering output for O_3 for last 15 days in Q_3 over 2015 to 2019.

summer, Q_3 autumn to Q_4 winter, with sudden high spikes are observed during the seasonal cycle changes mostly in the months of Jan, Mar, Jul, Sep and Dec, as analysed with the help of Figures[1]. These analyses devised that wind speed is positively correlated to NO_2 however, negatively correlated to O_3. In Q_1 from 3^{rd} to 7^{th}, and 26^{th} to 29^{th} in 2015 to 2019 there exists pattern of low speed winds. In Q_2 first fifteen days from 6^{th} to 12^{th}, and 27^{th} to 30^{th} wind speed keeps increasing and reached highest with volatile nature in years 2015 to 2019. Moreover, in Q_3 wind speed frequently changes from mild to increasing (also reached highest) magnitudes from 19^{th} to 20^{th}, and 26^{th} to 29^{th} in 2015 to 2019. In Q_4 from 8^{th} to 14^{th}, and 21^{st} to 25^{th} during 2015 to 2019 mild speed winds are measured. Furthermore, PM concentrations are more in Q_4 winter, and also in Mar, May, and Jul as

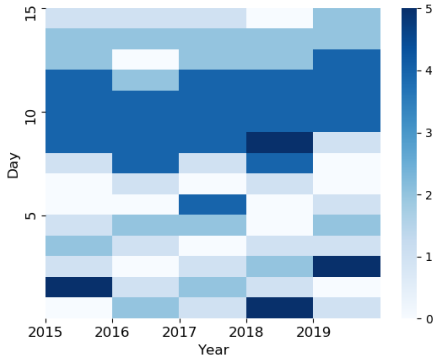


Figure 4.20: Clustering output for O_3 for first 15 days in Q_4 over 2015 to 2019.

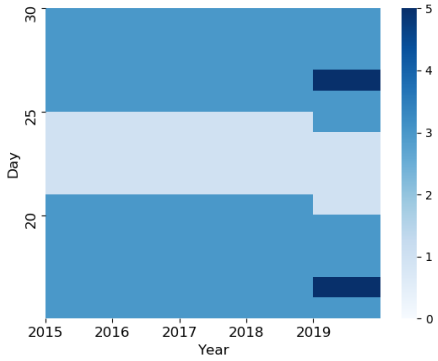


Figure 4.21: Clustering output for O_3 for last 15 days in Q_4 over 2015 to 2019.

concluded from the generated clustering outputs in Figures[1]. Analysing for Q_1 from 1^{st} to 15^{th}, and 23^{rd} to 27^{th} in years 2015 to 2017 represents existence of highest PM_{10} concentrations with constantly increasing level, however, with strong ban policies for diesel and old vehicles use by the German government and other regulatory movement restrictions and climate awareness, the PM concentrations are little controlled and reduced (comparison to earlier years) in 2018 and 2019. In Q_2 from 13^{th} to 15^{th} PM_{10} concentration increased in 2015 to 2016 and reached highest during years 2018 to 2019. From 27^{th} to 30^{th} high concentration was measured in 2015 to 2016 and then reduced to lowest in 2019 again during Q_2. Furthermore, in Q_3 first and last fifteen days of clustering output there exists frequently changing PM_{10} concentration from lowest to increasing in 27^{th} to 30^{th} with

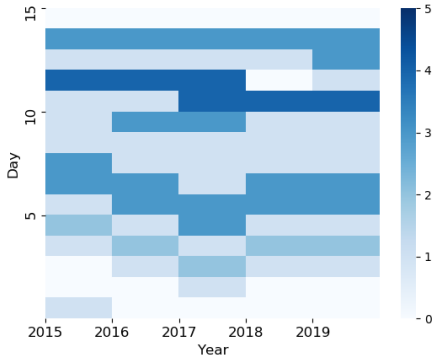


Figure 4.22: Clustering output for humidity for first 15 days in Q_1 over 2015 to 2019.

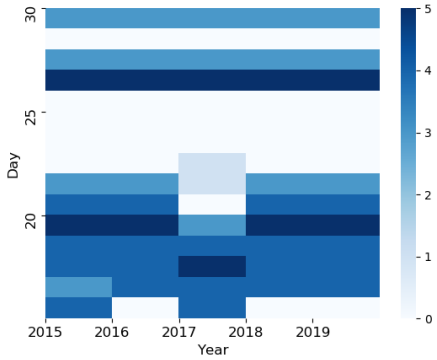


Figure 4.23: Clustering output for humidity for last 15 days in Q_1 over 2015 to 2019.

the concentration reaching highest during 2015 to 2019. In Q_4 from 2^{nd} to 4^{th}, and 13^{th} to 15^{th} in 2015 to 2019 PM_{10} concentration was measured highest. These interpretations (above analyses conclusions) provide a quick facts crosscheck supporting the present alarming air quality situation in the Stuttgart city and requirement of probable more control measures. In addition, the performed correlation analyses on pollution and meteorological datasets helped to uncover the important interrelationships, and also justified clustering analyses outcomes. Figure 4.12 contributes the following important points:

1. NO and NO_2 are 77% positively correlated to each other, with 27% positively correlated to $PM_{10\text{-}2.5}$, and negatively correlated to wind speed by 53%,

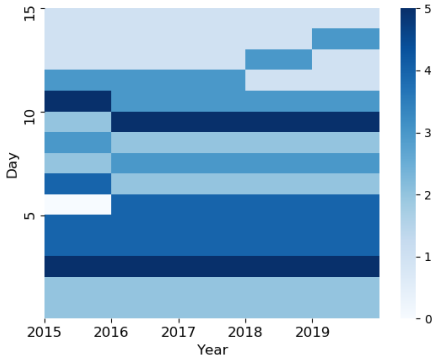


Figure 4.24: Clustering output for humidity for first 15 days in Q_2 over 2015 to 2019.

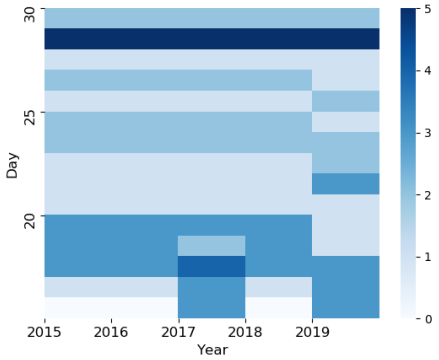


Figure 4.25: Clustering output for humidity for last 15 days in Q_2 over 2015 to 2019.

2. O_3 is 50% positively correlated to wind speed, 77% negatively correlated to NO and NO_2, and 27% negatively correlated to $PM_{10\text{-}2.5}$,
3. humidity is 27% positively correlated to NO and NO_2 and 50% negatively correlated to O_3,
4. wind speed is 27% negatively correlated to $PM_{10\text{-}2.5}$, moreover, PM_{10}, and $PM_{2.5}$ are positively correlated to each other with more than 87%.

Moreover, the developed seasonality analysis kit is used to provide interactive selections of considered meteorological and pollution parameters to analyse the concurred pattern in the dataset, in a time based frame over the years. Currently, the designed dashboard is in its first phase with

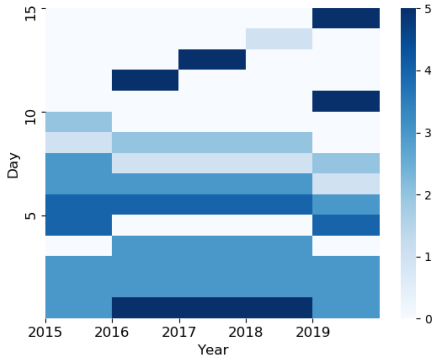


Figure 4.26: Clustering output for humidity for first 15 days in Q_3 over 2015 to 2019.

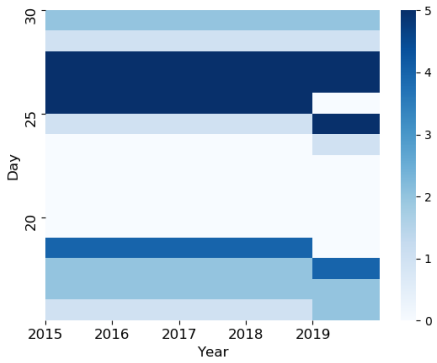


Figure 4.27: Clustering output for humidity for last 15 days in Q_3 over 2015 to 2019.

color based clustering display for each quarter over the years. This has helped in making the seasonality analyses tests easy, user interactive and comparable in the time domain.

4.1.5 Conclusion

The integration of new knowledge, innovative technologies in sustainable transformation is the motive of this work. The interpretations (above analyses conclusions) provide a quick facts crosscheck supporting the present alarming air quality situation in the city and requirement of probable more control measures. The interactive dashboard seasonality analysis kit of meteorological and pollution parameters would help to plan the future

environmental conditions. However, an improved approach is required that combines more environmental data, correlation analysis, temporal heat map and a better interactive visualisation integrating with the above developed ML visual predictors for multiple parameters in depth analysis for various time frames in a robust web platform.

4.2 Chapter Summary

This chapter is the first step of interaction with meteorological and pollution parameters visually. The first approach provides findings based on the temporal seasonality of hourly time series $PM_{2.5}$ and PM_{10}, NO, NO_2, and O_3 along with the measured wind flow and humidity. The temporal variations over the city centre in Stuttgart are analysed using an unsupervised approach to perform seasonal hierarchical clustering on a series of parameters NO, NO_2, O_3, PM_{10}, and $PM_{2.5}$, wind speed and humidity. Furthermore, the correlations between meteorological and pollution parameters clearly demonstrate the relationship between air pollutants, wind, humidity together in a combined temporal activities frame. Moreover, a dashboard is developed to provide the user desired time frame visualisation of these parameters. Further, a limitation in these methods is that they lack interactive comparison of the machine learning models simultaneously with their output display in one temporal visualisation query frame. This is an important requirement and is a motivation behind the visual assessment for meteorological and pollution parameters.

5

Air Quality Temporal Analyser & Geospatial Data Visual Assessments

This chapter presents air quality temporal analyser and geospatial data visual assessments designed in this work. The following section explains the setup of the developed visual assessments platform for air quality analyses and sensors health monitoring visualisation platform.

Developed Methods First the introduction of Air Quality Temporal Analyser (AQTA) is given in section 5.1, dataset used in subsection 5.1.1 and followed by approach in subsection 5.1.2. The results of AQTA are explained in subsection 5.1.3, followed by discussions in subsection 5.1.4 and conclusion in subsection 5.1.5. Later in the section 5.2 introduction of geospatial data and sensors health monitoring visual assessments platform

Parts of this chapter have previously been published in:

Harbola, S. and Coors, V. (2018), 'Geo-Visualisation and Visual Analytics for Smart Cities: A Survey', *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-4/W11, https://doi.org/10.5194/isprs-archives-XLII-4-W11-11-2018, 11–18;

Harbola, S. and Coors, V. (2019a), 'One Dimensional Convolutional Neural Network Architectures for Wind Prediction', *Energy Conversion and Management*, 195, https://doi.org/10.1016/j.enconman.2019.05.007, 70–75;

Harbola, S. and Coors, V. (2019b), 'Comparative analysis of LSTM, RF and SVM Architectures for Predicting Wind Nature for smart city planning', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-4/W9, https://doi.org/10.5194/isprs-annals-IV-4-W9-65-2019, 65–70;

Harbola, S., Koch, S., Ertl, T., and Coors, V. (2021a), 'Air Quality Temporal Analyser: Interactive temporal analyses with visual predictive assessments', *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, https://doi.org/10.2312/envirvis.20211083;

Harbola, S. and Coors, V. (2021b), 'An Interactive Platform For Environmental Sensors Data Analyses', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, VIII-4/W1-2021, https://doi.org/10.5194/isprs-annals-VIII-4-W1-2021-57-2021, 57–64.

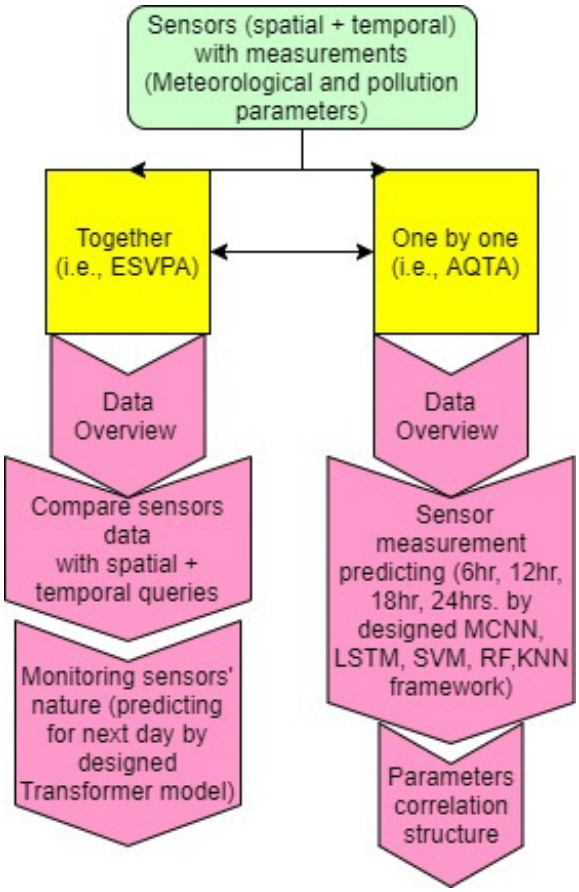


Figure 5.1: Algorithms comparative analysis flow chart.

is discussed, followed by methodology in subsection 5.2.1 and datset used in subsection 5.2.5. The subsection 5.2.6 explains the results of geospatial data and sensors health monitoring visual assessments platform followed by discussion in subsection 5.2.7 and findings, conclusions in subsection 5.2.8. Furthermore, the chapter summary is given at the end in section 5.3. Figure 5.1 provides a brief overview of this chapter workflow approach along with highlighting the motivation behind this work.

5.1 Overview: Air Quality Temporal Analyser

Visual data exploration often follows Shniderman's mantra (Shneiderman, 1996). The work related to visual prediction, time series visualisation and temporal analytical approaches which matches the keywords of the developed work were explored. Recent techniques (Badam et al., 2016; Krause et al., 2016) on visualising the time series data supported with

mathematical and statistical metrics enable the user to build reasoning about the considered temporal datasets interactively. Visualisation techniques, highlighting the anomalies and underlying trends correlations, through an undirected interactive search (Sacha et al., 2016) were developed. Moreover, time series visualisation were explored by providing examples of simple charts including stacked graphs, index charts, horizon graphs for visualising time series datasets. The representations of time series data become more contextual with the support of cluster, calendar-based and, spiral visualisations (Weber et al., 2011). More detailed and aggregated representations, using multi-resolution layouts for handling over-plotting in large time series datasets were developed (Fu, 2011; Hao et al., 2011). Moreover, they also reviewed the data mining method for classification, pattern exploration, segmentation and representation of time series data. Hochheiser and Shneiderman, invented dynamic query tools for time series dataset interactive explorations with user demand detailing (Hochheiser and Shneiderman, 2004). Chronolenses were proposed for time series data visual exploration and correlation analysis (Zhao et al., 2011a,b). Anomaly detection for modelling multiple time series (Chan and Mohoney, 2005), clustering and classification (Liao, 2005) techniques to identify the similarity of data patterns among time series dataset using weighted dynamic time warping (Jeong et al., 2011), distance metrics and agglomerative clustering have been developed (see subsection 4.1.1). Inter parameters relationships definition rules are revolutionised by Hetland and Saetrom (2005) with rule mining concept for time series database. The scientific temporal data visualisations are frequently used in support of interactive visual analytics and are well-accepted within the disciplines (Andrienko and Andrienko, 2003; Navarra et al., 2020).

Moreover, for understanding the temporal datasets and its trends, predicting future and patterns remains a very challenging task with a few interactive visual models and user explorations behaviour support. Predicting the time series data using statistical methodologies like regression analysis, and computational machine learning approaches like neural networks, multilayer perceptron, fuzzy logic and self organising maps have been successfully applied for the existing studies (Lorenc, 1986; Guilherme, 2007; Bollen et al., 2011; Venugopal et al., 2011). Visual prediction approaches in the act of visually predicting a time series variable by observing the predictions from a computational model, shown alongside with the time series representations for social media and financial datasets were designed (Hao et al., 2011; Lu et al., 2014; Badam et al., 2016). Furthermore, interaction techniques with engaging the user in an efficient dialogue in the contribution by people and computers to solve the task together $i.e.$, mixed-initiative interaction techniques have also been proposed (Horvitz, 1999,

2007; Endert et al., 2012; Kapoor et al., 2012). Data driven forecasting in visual predictions for time series dataset visualisation with highlighting the sequence and pattern in support of approaches to explore correlations in multivariate spatio-temporal data have been developed by Hao et al. (2011); Malik et al. (2012).

However, the increased usage of the environmental monitoring system and sensors installation on a day-to-day basis has provided more information in monitoring the current environmental conditions. Sensor networking advancement with quality and quantity for air parameters, has given rise to an increase in techniques and methodologies supporting temporal data interactive visualisation analyses (Hart, 2006; Bogue, 2008). Moreover, there exists a gap between the environment as observed and its digital representation in the user selected time frame for temporal data interactive analysis. Visualisation of meteorological and pollution data history and context plays an essential role in visual data mining, especially in exploring the large and complex datasets and environmental conditions. Including the context and historical information in the visualisation could improve user understanding of the environmental dataset exploration process and enhancing the re-usability of mining and managing techniques and parameters analysis to achieve the required insight. Moreover, traditional approaches cannot fully support the visual exploration of future trends in complex multivariate time series datasets such as weather, and healthcare, mainly due to their lack of consideration of inter-variable relationships ($e.g.$, if PM_{10} increases, NO_2 decreases). Exploring these relationships through "what if" questions ($e.g.$, what if PM_{10} increases?) could help the user to better judge the future environmental conditions than blindly trusting computational models that lack contextual information.

Thus, there is still a gap the user likely needs to bridge for comprehending the situation. The developed work overcomes these dissociations by proposing an Air Quality Temporal Analyser (AQTA), an interactive system-user interface for visual prediction of multivariate time series through deep learning models as well as interactive visualisation techniques for air quality parameters. Following are the contributions of the current work,

1. interactive temporal visualisation of historical, present and future data through various charts, to support the user in the interpretation of the data that may be useful for further stages of the mining process such as cluster identifications, important feature and pattern detection,

2. predicting the air quality standards for the desired temporal frame (dynamic) with five designed deep learning models, thereby highlighting the respective model's success and failure for inference

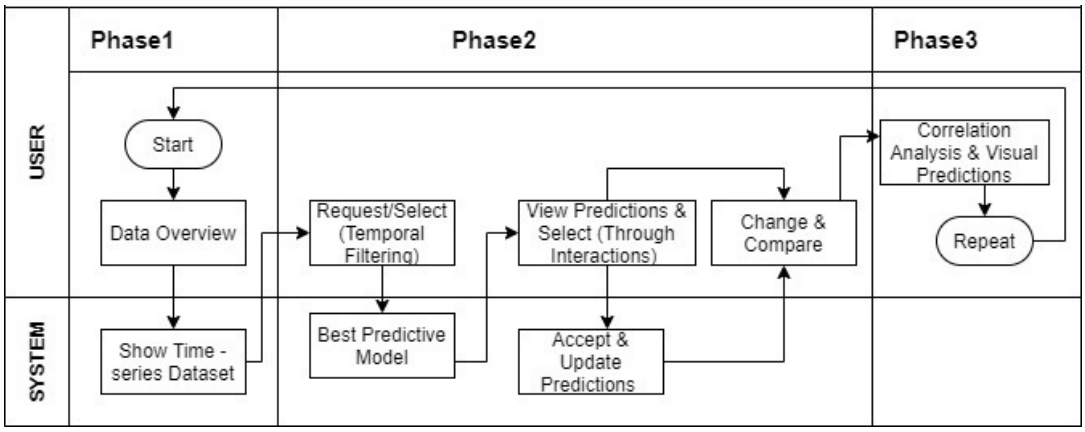


Figure 5.2: AQTA workflow maintains an interactive dialogue between user and the system for visual prediction and in depth analysis including correlation.

data along with supporting the arguments with easy graphical support and suggesting best option to choose,

3. visual preservation of context and historical information in all these user interactions.

These contributions combine together to form three phases (1-3 shown in Figure 5.2) of interactive AQTA with back-and-forth dialogues between user and AQTA. This interactive dialogue between the AQTA and the user continues until the user finds sufficient information to come to a conclusion. This would infer smart decisions for air quality planning, which in turn would help in proficient management and development of the city's resources. AQTA is validated for Stuttgart, Germany as a used case study. The remaining sections are organised as follows: system and datasets used and developed approaches are discussed in subsection 5.1.1 and subsection 5.1.2, respectively, subsection 5.1.3 and subsection 5.1.4 discuss the results, followed by conclusion in subsection 5.1.5.

5.1.1 Data Used

The temporal air quality datasets that are used and analysed in this study ("luftdaten selber messen" (Luftdata-se-Stuttgart, 2020)) provide city sensors measurements at several locations in Stuttgart. Historical datasets from 2016 to 2020 are measured at total 8 city centre locations with the wind (speed and directions), temperature, pressure and humidity along with NO, NO_2, O_3, PM_{10}, with temporal information attached in a

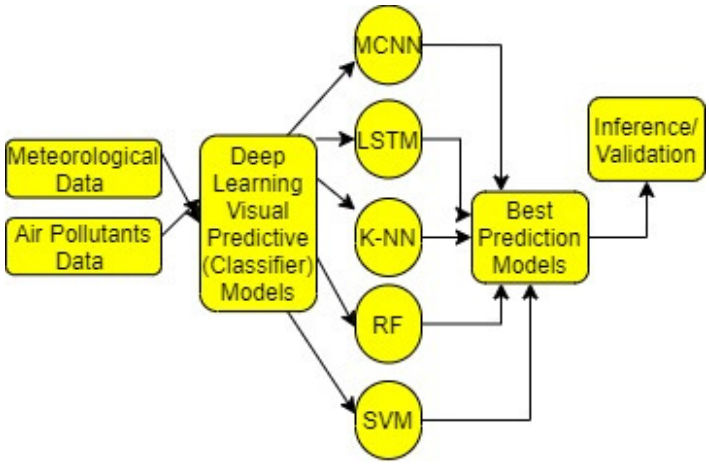


Figure 5.3: Predictive models analysis flowchart.

Class	Lower Range	Upper Range
calm	$min(parameter)$	$\mu - k_1\sigma$
light	$\mu - k_1\sigma$	$\mu - k_2\sigma$
mild	$\mu - k_2\sigma$	$\mu + k_2\sigma$
strong	$\mu + k_2\sigma$	$\mu + k_1\sigma$
strongest	$\mu + k_1\sigma$	$max(parameter)$

Figure 5.4: Various classes designed ranges.

30 minute time interval (Stadtklima-Stuttgart, 2021). The dataset of each location was separated into individual years for each parameter, using time information with past data first, followed by current data. This helps to perform an in depth study of air parameters. AQTA is implemented as a web based application using D3.js, Streamlit, Keras library (Chollet, 2017) with TensorFlow in the backend in Python and executed on Intel® Core ™ i7- 4770 CPU @3.40 GHz having four cores. Each designed air quality predictor (ML) model was executed separately for selected time series data for predicting the (dominating) class magnitudes and analysing air nature. The following subsection 5.1.2 explains the designed system architecture comprising models, graphs and database at the system side and interactive visualisation interface at the user side. Result in subsection 5.1.3 analyses and validates the outcome of sensor located at Stuttgart's city centre, and similar results were obtained for the other sensors as well. AQTA web deployment along with detailed figures are available in GitHub `//www.github.com/shharbola/EnvirVis_AQTA`.

5.1.2 Approach

The designed work combines different visual analysis of air quality parameters, integrated into AQTA platform. Figure 5.3 provides an overview of the workflow and highlights the motivation behind the comparative analysis of different models. Here, the time series air quality datasets comprise pollutants $i.e.$, PM_{10}, NO, NO_2, and O_3 and meteorological parameters like wind (speed and direction), pressure, temperature and humidity, with temporal resolution T and T_w ($w \rightarrow 1$ to m) denotes value of the selected parameter (above mentioned) at time w, where 1 and m are the first and last values in the dataset, respectively.

Air Quality Predictor

Multiple samples are designed using the dataset for training and testing the developed prediction algorithms. A sample consists of a feature vector as an input with a corresponding output class. $Real_{V_b}$ (a scalar) consecutive values of considered parameter, from T_w to $T_{w+Real_{V_b}}$ form a feature vector of dimension $Real_{V_b} \times 1$ which is the input of the sample. $Real_{V_f}$ (a scalar) successive values of selected parameter after the last value in the input $i.e.$, $T_{w+Real_{V_b}}$, are used to define the sample's output class. Mean (μ), and standard deviation (σ) of the parameter of the entire dataset are calculated. Various class boundaries are designed using μ and σ as shown in Figure 5.4.

Among $Real_{V_f}$, count of values occurring in each class in Figure 5.4 is noted, and the class that has a maximum count $i.e.$, dominant, is assigned to the sample. Similarly, multiple samples based on the selected parameter are created by taking $Real_{V_b}$ values in the corresponding input from T_w to $T_{w+Real_{V_b}}$ by varying w from 1 to m - $Real_{V_f}$, at an increment of 1. The outputs of these samples are designed as discussed above. Likewise, samples based on other parameters (each independently) are created for each dynamically selected parameter as discussed above. Thus, at this stage, for $Real_{V_b}$ values in the input from T_w to $T_{w+Real_{V_b}}$, there would be nine sets of samples, based on PM_{10}, NO, NO_2, O_3 and wind (speed and direction), pressure, temperature, and humidity. Here, in this analysis the size of $Real_{V_b}$ and $Real_{V_f}$ are kept equal with four user options,

1. 12 representing 6 hours as temporal resolution of considered dataset is 30 minutes,
2. 24 representing 12 hours,
3. 36 representing 18 hours, and
4. 48 representing 24 hours

. These conditions ensured comprehensive and accurate analysis of the data with respect to independent and different user selections.

The first developed air quality predictor ML model is Multi-Convolutional Neural Network (MCNN) that has five single CNN, say (CNN_1, CNN_2, CNN_3, CNN_4, CNN_5). Each of these CNN_i ($i \rightarrow 1$ to 5) has its own input layer, three consecutive 1D convolutional layers and last convolutional layer of each CNN connects to a common fully connected layer which is followed by another fully connected layer and an output layer. The architecture is explained in detail in subsection 3.1.1, subsection 3.2.1. MCNN and 1DM are the same architectures. The output layer is a softmax layer (Su et al., 2015), with the number of neurons same as the number of the classes. There are five classes in the present study as shown in Figure 5.4. The MCNN is trained and tested separately for the prediction of dominant temporal nature of the selected parameter (PM_{10}, NO, NO_2, O_3, wind, pressure, temperature and humidity). Therefore, for an inference sample, the MCNN could predict the air quality parameters classes separately and visually highlight time series data recurring motif.

The developed Long Short Term Memory (LSTM) model (second) is a special kind of Recurrent Neural Networks (RNN) capable of learning long term dependencies with a chain like structure. This has an input layer, four neural layers ($NL1$, $NL2$, $NL3$, $NL4$), $i.e.$, three sigmoid layers supported with two tanh layers and an output layer. The architecture is explained in detail in subsection 3.2.1, subsection 3.1.1. The input layer is One Dimensional (1D) of the size of $Real_{V_b}$. The output layer is a softmax layer, having the number of neurons the same as the number of the classes $i.e.$, five.

The third designed time series prediction model uses K-Nearest Neighbors (KNN) which is a supervised classification algorithm. KNN based method makes predictions on the fly by calculating the similarity between an input observation and values in the dataset, with respect to time. Here K value is decided empirically and kept fixed in all parameter analysis. The designed SVM based predictive fourth model classifies the data by finding the best hyper-plane that separates all data points of one class from those of the other class. The best hyper-plane signifies the one with the largest margin between the classes. Similarly the last designed Random Forest (RF) based model uses a decision tree as a decision support tool for classification. When the RF is given a training sample, it formulates a set of rules which are used to perform predictions. Moreover, RF uses sufficient decision trees, to ensure the classifier does not over-fit the model. The advantage of the RF as a classifier is that it can handle missing values, and the classifier could be modeled for categorical values. Therefore LSTM, MCNN, SVM, K-NN and RF (five ML models) are used to predict

meteorological and pollution parameters separately.

During training, the sample's feature vector of dimension $RealV_b \times 1$, forms the input of the designed models, while the sample's output class forms the output of these models. The objective behind using a variety of supervised prediction models is to provide a possible option of selecting models based on best (compare) accuracy with respect to the various date, time, and parameters conditions. The previous paragraphs discuss the various developed models of temporal air quality prediction. Besides prediction, the detailed analysis of historical air quality parameters are also performed in this work. Temporal filtering using Pearson correlation method help to derive the relationships along with highlighting interconnections between the meteorological and air pollutants. The user could select the parameters over the desired time frame and compare the patterns interactively in AQTA, thus, making the analysis more diverse and refined.

Visual Interaction Design

AQTA besides being air quality predictor, also provides tooltiping, brushing and linking for maintaining the transparency and combining different visualisation methods between user-computer dialogue efficiently and preserving the working memory of the user during interactions (Shneiderman, 1996; Horvitz, 1999). Figure 5.2 provides an overview of AQTA workflow (phases 1-3), with highlighting the system-user interfaces of visual predictions comparative analysis. **System:** consists of historical air quality temporal database, trained ML models, structure of various graphs and charts, and accepts user queries. **User:** interacts with this system in various ways. The user selects, inspects and views the states of the parameters with past present and future (predictions) information. The user could also choose among different ML models with analysing the performance of each selected model (MCNN, LSTM, RF, K-NN and SVM) in terms of total accuracy and difference metrics incorporated with the interactive display through various graphs and charts. The user could change the time step allowing for a different prediction duration, and compare the results with the time series dataset and the outcome of each model. This allows the user to decide which prediction algorithms are the best and provides sufficient information to make a decision.

The system works as per user desires with additional information of revelling the correlation among the selected parameters and answering "what if" questions of nine parameters dependency with each other within selected time frame. Furthermore, detailed analysis of the patterns in the dataset in the three phases of AQTA are carried out using additional charts, heat-map, time histogram, that are explained below.

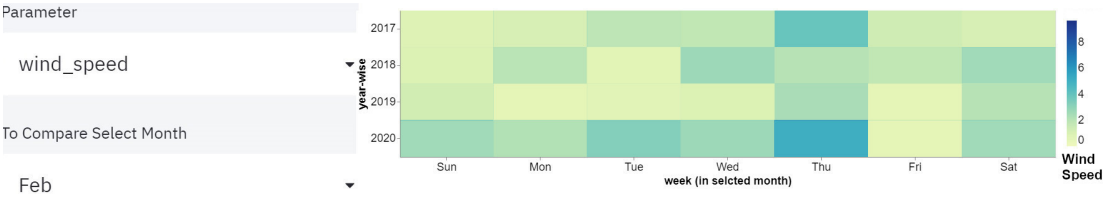


Figure 5.5: Data inspection (a part of Phase 1) week-wise over the years for selected parameter.

Inspecting Data History Visualisation (Phase 1) The phase 1 visualisation of AQTA uses time series stack chart with calendar heat-map to provide interaction with the air quality datasets visually. The inspecting overview shows the overall patterns for multiple parameters of interest selected from air quality parameters list available in the interface (Figure 5.5). The time dataset overview design contains horizon graphs. The effective discrimination option in horizon graphs makes it more desirable (Javed et al., 2010). This is accompanied with stacked chart to provide a detailed time series data inspection of parameters magnitudes with calendar heat-map view option in order to compare the trends among air quality parameters based on the months during a year. The user could select each year and then even explore in detail for each day with 30 minutes (here the sensors' data temporal resolution) for air quality parameters temporal analyses. This phase provides a detailed understanding of the air quality data history and preset with highlighting the patterns which are actually present and measured by the sensors (here no smoothing or data cleaning is performed $i.e.$, real original datasets).

Prediction Visualisation (Phase 2) The phase 2 visualisation consists of square-time charts, and temporal circle mark chart coupled with histogram highlighting the predicted value (Figure 5.6). Predicted outcome with respect to time frame (6hr, 12hr, 18hr, 24hr) choices are displayed with the help of square-time chart with tooltip highlighting the class assigned and color encoding makes it easy to distinguish in detail the classes with respect to each predicted value in the time frame. Each class is assigned dynamic color encoding according to predicted class range. The comparison and performing the analysis of predicted versus the actual values is shown with the help of time series square-time graph with the color encoding representing the difference of actual and predicted (Figure 5.6 (a)), that occur in the range $i.e.$, (-4, -3, -2, -1, 0, 1, 2, 3, 4) calculated by assigning 1 = calm, 2 = light, 3 = mild, 4 = strong, 5 = strongest as in Figure 5.4.

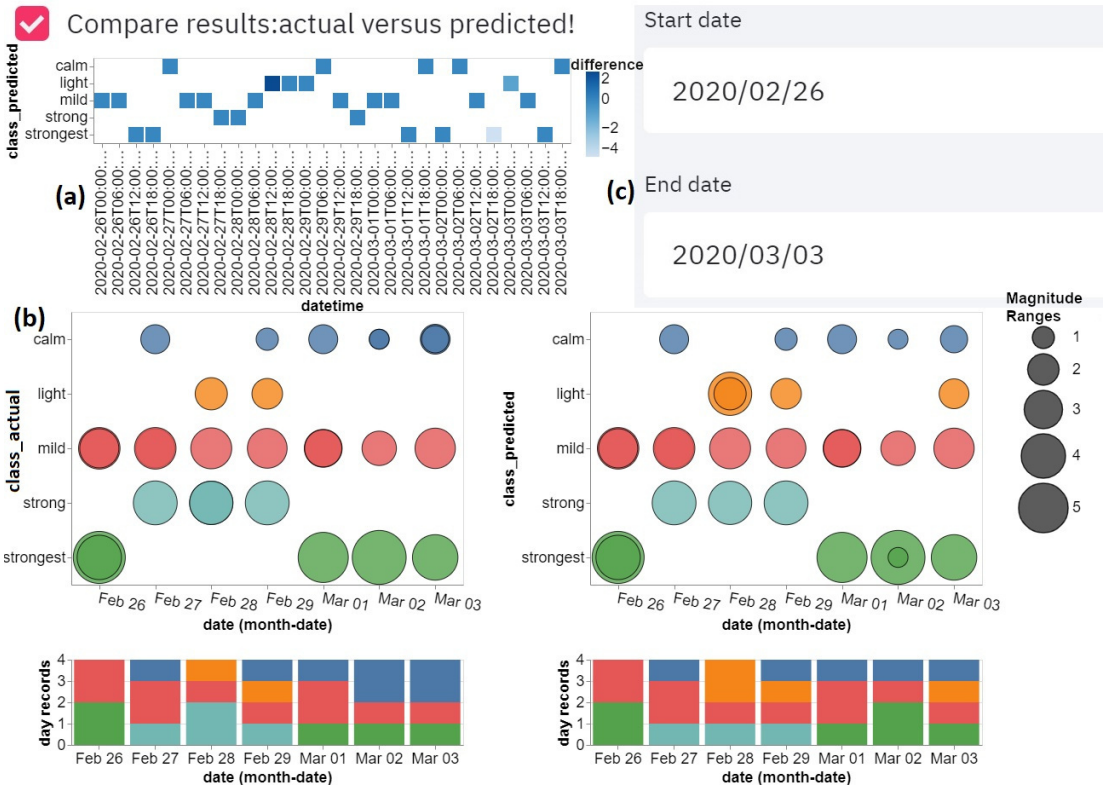


Figure 5.6: Phase 2 inference, comparing actual versus predicted output.

Tooltiping is also added to this representation to make it easier for user to understand the actual and predicted values along with their respective difference in the time frame.

In order to provide a detailed comparison and more easy interaction by double encoding, mark circle with integrated histogram graph is designed (Figure 5.6 (b)). Here the circle radius is governed by the class ranges and color according to the assigned class with respect to time. The histogram shows the count of the records estimated or predicted each day and binned according to the assigned class patterns. Both actual (Figure 5.6 (b) left) and predicted (Figure 5.6 (b) right) values are compared in this interface with clearly highlighting the pattern of meteorological and pollution parameters in time frame, which helps user to make advance and comparative estimation of the environment and its pattern with model's success information.

Correlation Visualisation (Phase 3) The phase 3 visualisation of AQTA is implemented as an air quality parameters' correlation structure detailed analyses. The time series exploratory analysis of meteorological and pollu-

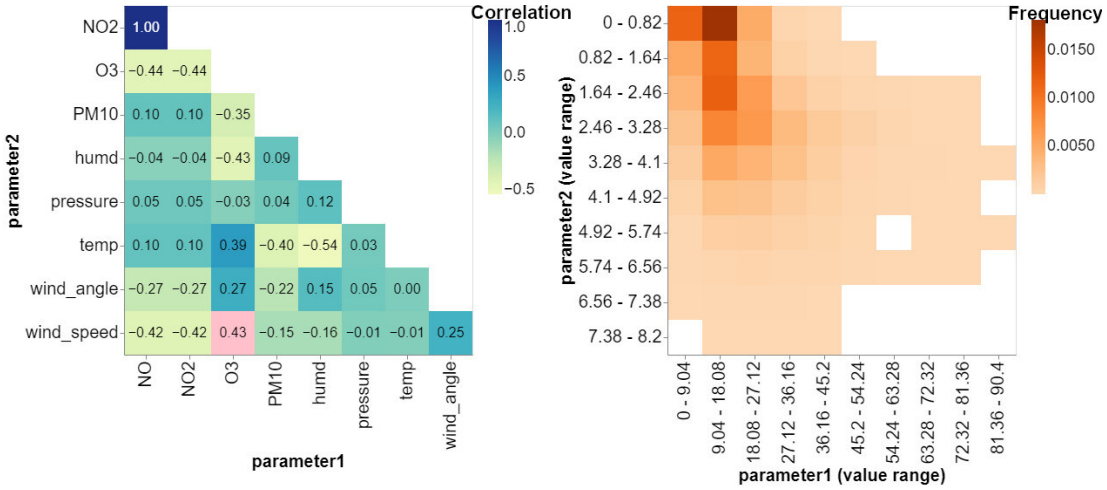


Figure 5.7: Temporal visual correlation analysis using correlation heat-map (left) linked with 2D histogram (right).

tion parameters also requires supporting, identifying the correlation and how these parameters are controlling and effecting each other's nature interaction. Pearson correlation method is used for analysing these relationships among the parameters. The correlation graph (Figure 5.7) explores the correlation structure of the meteorological and pollution parameters dataset using two connected subplots: an interactive correlation heat-map (Figure 5.7, left image) and a 2D histogram showing the density of values (Figure 5.7, right image).

Clicking on a cell in the correlation heat-map shows correlation coefficient value for that particular cell, (shown in pink highlight in Figure 5.7, left image), where parameter1 (on X-axis) and parameter2 (on Y-axis) represent associated air quality parameters of the selected cell on X-axis and Y-axis. Selection linked binning on the fly is performed for the selected cell generating a 2D histogram (detailed bins) between parameter1 and parameter2, with the advantage of highlighting overlapping values leading to a higher density of values (frequency) in the darker color bins, making it clear to the observer that there are more similar range values in the selection. Thus, the correlation heat-map shows the parameters and value ranges (all) associated with that particular cell and the corresponding data in the 2D histogram. This enables user to quickly see the pattern in correlations using the heat-map, and allows to zoom in on the dataset underlying those correlations in the 2D histogram. All the graphs and subsections integrated with visual predictions help the user to provide a more clarity of the time series and environmental conditions. AQTA (1-3 phases) tries to

bring all the information together that could be derived from air quality parameters condition in the considered city (Stuttgart, Germany) in order to derive the time series patterns and correlations.

5.1.3 Results: Use Case

AQTA has been evaluated using Stuttgart (Germany) city air pollutants, $i.e.$, PM_{10}, NO, NO_2, O_3 and meteorological parameters like pressure, temperature, wind and humidity. The initial findings are presented that corroborate the city's COVID lockdown (year 2020) conditions and sudden changes in patterns, highlighting the improvements in the pollutants concentrations. AQTA was used for visual analysis of Stuttgart's COVID lockdown air quality situation (in year 2020) to facilitate visual exploration of prediction models outcome and reality conditions that occurred during this sudden pandemic. AQTA results were compared with real world measurements to support analyser inference outcomes and interaction in subsection 5.1.3, followed by subsection 5.1.4 for discussion.

Inference

Several samples, each having input and corresponding output, were created as described in section 5.1.2. Values of k_1 and k_2 (Figure 5.4) were empirically taken as 0.80 and 0.50 respectively (same for all parameters), so that a sufficient number of samples occur in each class. Moreover, Synthetic Minority Oversampling Technique (SMOTE) was used to do up-sampling of the classes having less number of samples. Total samples for a given year were randomly split into training and testing with 35% of the total samples as the testing samples. The designed models were trained and tested on these samples. When samples were prepared for the inference (validation) for year (2020), the samples were created similar to model training and testing phase (as mentioned in subsection 5.1.2). The models had never seen the dataset which were used in inference therefore the pattern and class predicted dynamically, were predicted based on the designed models achieved accuracy. The obtained accuracies for five designed models are approximately between 90% to 95% (see subsection 3.1.1, subsection 3.2.1).

These classification outputs are shown in the supplemental material on Github. These outcomes represent AQTA phase 2 of Figure 5.2. The classes square chart uses diverse color coding to highlight the model's predicted classes assigned with respect to selected time frame of (6hrs, 12hrs, 18hrs and 24hrs) future prediction. Class specific color coding provides more distinguish representation irrespective of the selected time frame (small or large), that helps in quick user understanding and assessment of a lot of

predicted information at one go. The graphs (Figure 5.6) comparing the actual and predicted results difference, highlight the success and failure of the selected predicted models in the selected time frame (as shown in Figure 5.6 (c)). The difference (actual - predicted) of the selected model classification outcome is shown with square chart (Figure 5.6 (a)), here sequential single-hue schemes (blues) encoding shows the difference values $i.e.$, (-4 (light blue), < 0 = model success, < 4 (dark blue)) attached with tooltip information. Another graph, circle mark charts, represents actual and predicted classes separately (Figure 5.6 (b)). In these circle mark charts, the radius encodes the ranges of the assigned classes (calm < light < mild < strong < strongest). Integrated histograms at the bottom of these graphs denote each day's (overall) predicted and actual record of classification outcome, with colors and conditional selection are linked with the above circle mark charts. The together build selection between circle mark charts and corresponding histograms, gives the user option to filter the outcome as per the requirements. This helps in detailed analysis of the actual and predicted classification outcomes and model's success-failure overview, in each selected time frame and arriving at a conclusion to pick the best model.

Interactions

The data inspection $i.e.$, phase 1 is used to provide user the freedom to visually analyse all the historical data (available in database) with graphs by temporal queries. The options available for user are either to compare all the years with respect to month, day for the desired parameter or to explore in depth each year independently with querying based on week (Figure 5.5), date and time with overall option palette available to change, update the selection, process new one, save the results and return. Therefore, users can use controls, which provide zooming, selection, tooltiping and saving outcome (image format) options, to view the models classification distributions at different time frame. The user can use several available options on the screen, to get back to the default views, change the selection, reset the main phase view, the phase details views, or all the views.

The output of phase 3 (Figure 5.7) is the temporal correlation analysis of meteorological and pollution parameters with yearly selection option available to user. Creating one cohesive interactive plot using correlation heat-map linked with 2D histogram (showing the density of values), helps to answer queries related to parameters interrelationships and how their dependency fluctuates in time with comparison option. Binning on the fly, with user parameters selections and displaying the correlation (heat-map with yellow green blue sequential multi-hue schemes) and frequency (2D

histogram with oranges sequential single-hue schemes) allow to have details of an individual correlation as shown in Figure 5.7. The interactive chart enables to quickly distinguish pattern in correlations using the heatmap, and allows to zoom in on the meteorological and air pollution data underlying those correlations in the 2D histogram. This indicates that the correlation leans heavily on the tail of the data and vice versa. Visual correlation analysis queries would help to understand the data and temporal dependencies more clearly with interactive charts that make understanding very easy and less time taking, making environmental planning more comprehensive and interesting.

5.1.4 Discussion

ML based prediction algorithms used in AQTA are described in detail in subsection 3.1.1, subsection 3.2.1, section 2.3. These approaches with good prediction results are applied in phase 2 of AQTA to achieve an interactive visual prediction, and pattern analysis platform. This aids user to understand easily the insight of data, complexity of the parameters, trends and details, and air quality impact. AQTA focuses on integrating and linking the simple charts representation to discover complex air quality parameters interactively in various time frames, with options to have a visual data overview (history and present in phase 1), predicting future with model success, failure comparison (phase 2), and a correlation structure of their interrelationships (phase 3).

The designed framework is successfully implemented for the Stuttgart city central location. However, it could be applied to any number of sensors for any given location (area) with some ML tuning and training of the respective datasets. The air pollution from predominantly non-traffic-related pollutants ($e.g.$, dust deposits) has decreased significantly in recent years. The traffic-related pollutants ($e.g.$, NO, NO_2, PM_{10}, O_3) remain at a high level in the city (Stadtentwicklung.berlin.de, 2021). The city's air quality is controlled and not deteriorating further, due to the strong monitoring and control measures by the state governments, city's policymakers and increased environmental awareness among people. But still the AQTA analysis shows that during summer and autumn of the year 2019, PM_{10} trends are alike as in the previous years 2017 to 2018 with a few reductions. Furthermore, there is depletion in PM_{10} concentrations during the summer and autumn of the year 2020 probably due to the strict lockdown and movements restrictions. However, the decrease in the concentration of one parameter and increase in others does not ensure that the overall air quality is improved $e.g.$, PM_{10} is observed reduced in Oct 2020, while O_3 concentration is higher. The reasons behind these relationships and

trends are more evident with the correlation structure integrated with this analysis, highlighting that PM_{10} is positively correlated with NO and NO_2, while negatively correlated with O_3. Similarly, NO and NO_2 are negatively correlated with O_3. Thus AQTA allows the actual data to convey itself and used to upgrade the user's hypothesis with the best understanding.

PM_{10} concentrations were predicted for 22-29 March 2020 when there were strict COVID lockdown restrictions during these days. The ML models predicted the air quality parameters with good accuracy during these conditions. Thus, the developed AQTA framework has good potential for visual analytics along with prediction in different conditions. While analysing parameters from 20 April 2020 to 1 May 2020 and taking 6 hours time in future, LSTM model predicted the NO and NO_2 concentration to be strongest on days 22, 28, and 29 April 2020. It was between calm and mild for rest of the days. The comparison of predicted values with the real data showed approximately 95% accuracy of the model. When the relaxation in the lockdown was given one month later, at that time also, the model gave good results. Further, the model also predicted the pressure range from calm to mild on 21 April, strongest class on 23 April, and calm on 28 April. The predicted and actual values matched for 23, 28 April but there was a mismatch for 21 April. As pressure and PM_{10} are positively correlated their spikes and patterns show similarities with their effects over the days which also cross validates the correlation with reference to data range trends. PM_{10} concentrations were observed to be higher specially on Fridays (apart from other weekdays) in February of year 2020, as well as on Fridays and Saturdays in April and on Saturdays in May of 2020. Similar trends were observed for NO and NO_2 concentrations during the same time frames (correlation discussed above). Usually these trends were also similar to previous years, weeks, and days patterns, with only fluctuation in concentrations ranges (calm to strongest). These patterns could be because people might be using public transports and shared cabs on working days. Transportation emits more than half of NO and NO_2 in the air. During the weekend, people travel to their homes, have family outings besides other important travel plans, thereby contributing to higher pollutants concentrations. PM_{10} concentrations predicted using LSTM for new year eve's on 31.12.2020 to 01.01.2021 from 12:00 pm to 12:00 pm were between mild to strong, then to strongest (12:00 am to 2:00 am) and then mild ranges which matches with the published report of Stadtklima Stuttgart on 01.01.2021 on PM_{10} concentrations in Stuttgart on New Year's Eve 2020-2021 (Stadtklima-News, 2021).

The day wise analysis of wind speed was also performed for February 2020. It was observed that on Thursday the magnitude of wind speed occurred mainly between strong and strongest classes. This trend was also

noticed for the previous years as well. Similarly for temperature on Monday (strongest), Saturday (strong), Sunday (strong), and Friday (mild) classes patterns occurred within the selected time frames. Such analysis helps in proper utilisation and planning of renewable sources like wind. Moreover, wind speed, pressure and temperature are positively correlated to each other, while wind direction and speed are in positive interrelationship with O_3. Therefore, it was also observed that the local winds could often develop that do not cause high magnitude winds, but play an essential role in local ventilation of the city areas and determine the spread of air pollutants (as found from correlations insight discussed above). The Stuttgart region is one of the areas with the lowest rainfall in Germany, mainly due to the lee location (Black Forest, Swabian Alb) and precipitation conditions playing a significant role in cleaning the atmosphere through the wet deposition. Moreover, the humidity of an area is highly controlled by the wind directions as they are positively correlated. In the year 2020 April, May and August months, the measured humidity is lower (on average) in comparison to the same months in 2019. These trends also matches with the changing wind directions occurred during the same year and months patterns. Due to the high temperatures trends in recent years, combined with the existing humidity patterns, Stuttgart is one of the areas with increased heat load (approximately 30 days), with occasional cold fillips and this infer seems coherent with the state climate published annual report (Stadtklima-News, 2021). Hence, AQTA provides an add-on to the existing literature in terms of air quality multiple time series datasets dynamic visual predictions along with its detailed analyses, comparisons and validation with reality.

5.1.5 Conclusion

This work presents Air Quality Temporal Analyser (AQTA), an interactive system to support visual analyses of air quality data with time. This interactive AQTA allows the seamless integration of predictive models and detailed patterns analyses. While the previous approaches lack predictive air quality options, this interface provides back-and-forth dialogue with the designed multiple Machine Learning (ML) models and comparisons for better visual predictive assessments. These models can be dynamically selected in real-time, and the user could visually compare the results in different time conditions for the chosen parameters. Moreover, AQTA provides data selection, display, visualisation of past, present, future (prediction) and correlation structure among air parameters, highlighting the predictive models effectiveness. AQTA has been evaluated using Stuttgart (Germany) city air pollutants, $i.e.$, PM_{10}, NO, NO_2, O_3 and meteorological parameters like pressure, temperature, wind and humidity. The initial findings are

presented that corroborate the city's COVID lockdown (year 2020) conditions and sudden changes in patterns, highlighting the improvements in the pollutants concentrations. AQTA, thus, successfully discovers temporal relationships among complex air quality data, interactively in different time frames, by harnessing the user's knowledge of factors influencing the past, present and future behavior, with the aid of ML models. Further, this study also reveals that the decrease in the concentration of one pollutant does not ensure that the surrounding air quality would improve as other factors are interrelated. The AQTA can be further advanced by highlighting the locations of different sensors with an add-on to the sensor nature's monitoring and, this motivates the following technique.

5.2 Overview: Geospatial Data Visual Assessments

The cities generate and store a lot of spatio-temporal data along with environmental parameters, constantly using various environmental monitoring sensors. Moreover, keeping track of the surroundings, and managing spatial data includes cities, rivers, roads, and countries with increasing demand for environmental monitoring, smart cities planning and resource management. The development and industrial advancement for uplifting human standards have contributed to a comfortable life on one hand while consequences of environmental changes, and pollution on another. Chimneys' discharge, waste from industries, vehicle smokes, and construction sites release consist of tiny air pollutants that upon inhalation, cause respiratory problems, lung and heart diseases. Therefore, meteorological parameters $i.e.$, humidity, wind (speed and direction) along with air pollutants like $PM_{2.5}$ and PM_{10} require regular monitoring. The surrounding air quality and well being fluctuate with these parameters atmospheric concentrations (Chen and Zhao, 2011). The increased levels of pollution parameters are due to regional transportation and local emission sources in the developed areas (Chen and Zhao, 2011; Jasen et al., 2013). One of the primary sources of particular matters is regional transportation comprising diesel vehicles that contribute significantly to pollution (Wallace and Hobbs, 1977). Moreover, sensors' (spatial and temporal) data is a combination of the georeferenced geographical entity presented by the attribute, location, and time as continuous, more extensive size data. Data Visualisation (Vis) is the practice of translating information into a visual context. Moreover, Visual Analytics (VA) is a sub-field of Vis which integrates data analyses with highly interactive visualisations. Furthermore, in the scientific domain, a visualisation setting would help bring the geospatial data clarity by displaying the patterns and variations that mostly remain undiscovered

in the theoretical or text data (Sun et al., 2013; Sun and Li, 2016). Some existing studies have been performed to infer the seasonality and patterns insight for meteorological and pollution parameters independently (see subsection 4.1.1). Integrating interactive Vis techniques help in representing the geospatial data and attached environmental information together in one frame. In this a more dedicated version using interactive charts, maps, and graphs to deliver the relationship trends between the parameters are presented, which are usually missing in traditional static charts, spreadsheets, and files (Horvitz, 2007; Aigner, 2013; Liu et al., 2017). However, VA is a combination of interactive Vis and automated analyses techniques. This supports the easy interpretation of spatio-temporal data and provides a better understanding of making choice potential by splitting a city environmental data into multiple parts ranging over time, space and several spatial scales (Kurkcu et al., 2017; Stratigea et al., 2017). Data filtering and smoothing methods are applied to the data in most of the above considered pieces of literature. These adjustments modify the most temporal dataset originality. Interactively visualising the sensors and their data measurements concerning the time frame helps monitor these parameters. A thorough study of the meteorological parameters, their trends, including their impact on $PM_{10\text{-}2.5}$ understanding, could be helpful.

The above research suggests that several questions remain to be addressed, such as temporal wind variations, $PM_{10\text{-}2.5}$ concentration fluctuations and in connection with user desired time frame, without modifying the authenticity of the original temporal dataset. An interactive system, AQTA, is developed, supporting the visual analyses of air quality data with time (see subsection 5.1.2). It discovers temporal relationships among complex air quality data, interactively in different time frames, by harnessing the user's knowledge of factors influencing the behaviour with the aid of ML models, but on a small scale, focusing on each sensor (individually) while missing the attached spatial knowledge. A more refined understanding of volatile sensor's measurements temporally which would increase the user interaction with recorded measurements and spatio-temporal information, is still required. This motivates the current research. The climate fluctuation and meteorological data monitoring concerns increased the demand for such a web interface to study the measured data history interactively along with the sensors' nature monitoring for the future. This idea is implemented and expanded as a case study for Stuttgart (Germany).

Thus, unsupervised Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) based clustering algorithm and ML model using Transformers Network are designed for sensors nature monitoring and highlight the dominating sensors locations working on the original temporal datasets by taking into consideration the above listed gaps, and

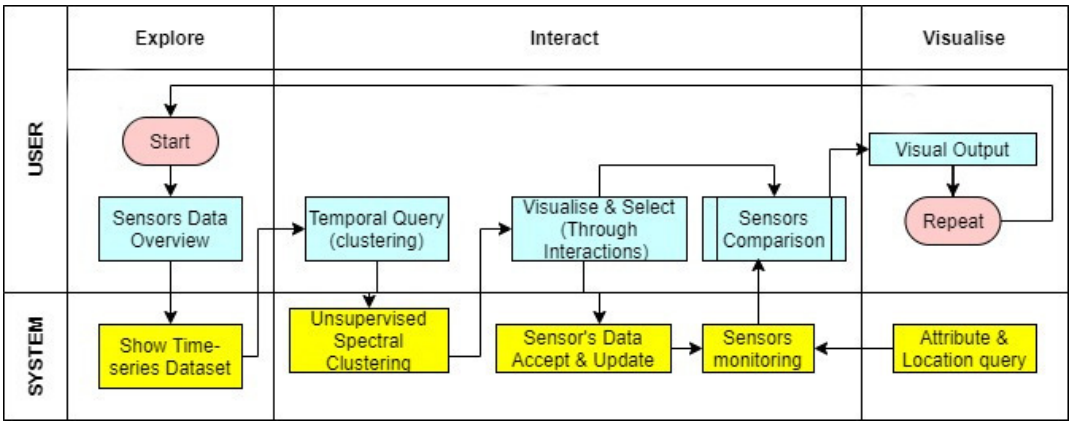


Figure 5.8: Sensors visual comparative analyses flowchart.

provide solutions of these gaps. These frameworks combine together to form (shown in Figure 5.8) Environmental Sensors Visual Prediction Assessment (ESVPA) an interactive visualisation platform with user choice of parameters selection, delivering temporal variations of spatio-temporal information. Therefore, the current study proposes HDBSCAN clustering and sensors nature monitoring queries with the following contributions:

1. interactive temporal visualisation of unsupervised cluster identifications to support the user in the interpretation of the meteorological and pollution parameters,

2. predicting sensor nature using Transformers Network, supported with visualisation of designed model dynamic training, testing and accuracy metrics assessments. This helps in highlighting the respective model's success and failure for inference data,

3. visual preservation of spatial, non-spatial context and historical dataset information on user selected temporal frame, and

4. unboxing the complexities of ML design with visualisation thus, making the concept understanding more explainable and straightforward.

This interactive visualisation platform would help to infer smart decisions for surrounding quality planning, which would increase the ability to devise more green ideas for city's resources, innovative development and management. The remaining contents are organised as follows: section 5.2.1, 5.2.5, 5.2.6, and 5.2.8 present methodology, datasets, results, discussion and conclusion, respectively.

Table 5.1: Various classes designed ranges.

Class	Lower Range	Upper Range
1	$min(selected_{parameter})$	$\mu - k_1\sigma$
2	$\mu - k_1\sigma$	$\mu + k_2\sigma$
3	$\mu + k_2\sigma$	$max(selected_{parameter})$

5.2.1 Approach

The developed interactive web interface provides a platform to view and analyse in detail several sensors and their measurements in Stuttgart city along with spatial and temporal information. Each of the sensors are measuring parameters like $PM_{2.5}$ and PM_{10}, humidity, wind (speed and direction). The section 5.2.2 explains the developed system architecture comprising unsupervised HDBSCAN clustering, sensors nature prediction using Transformers Network is discussed in 5.2.3, and interactive visualisation platform insight is described in 5.2.4.

5.2.2 Unsupervised HDBSCAN Clustering

All sensors time series measurements (for each sensor location) are studied using unsupervised clustering and sensors' location queries. Each parameter's values are normalised, and then temporal filtering is applied. The standard deviation and mean of the parameter's values are computed. The normalised values are calculated by subtracting the parameter's values from the mean, and the resultant values are then divided by the standard deviation. In the present work, temporal filtering is applied based on the user's choice. These user selection temporal query division helps in detailed analysis of the considered parameters as per user desires. HDBSCAN is applied in this study on sensors' measurements with noise which is an extension of Density-Based Spatial Clustering (DBSCAN) by converting it into a hierarchical clustering algorithm. It performs DBSCAN over varying epsilon (eps) values ($i.e.$, eps-neighborhood of point X, defining the radius of neighborhood around a point X) and integrates the results to find a clustering that gives the best stability over eps (Campello et al., 2013). This allows HDBSCAN to find clusters of varying densities (unlike DBSCAN). Therefore, for historical data HDBSCAN returns good clusters with little parameter tuning. The minimum cluster size parameter is intuitive and decided empirically in this study. Values of k_1 and k_2 (Table 5.1) were empirically taken as 0.75 and 0.35 respectively (same for all the parameters), so that a sufficient number of samples occur in each class (as shown in Table 5.1).

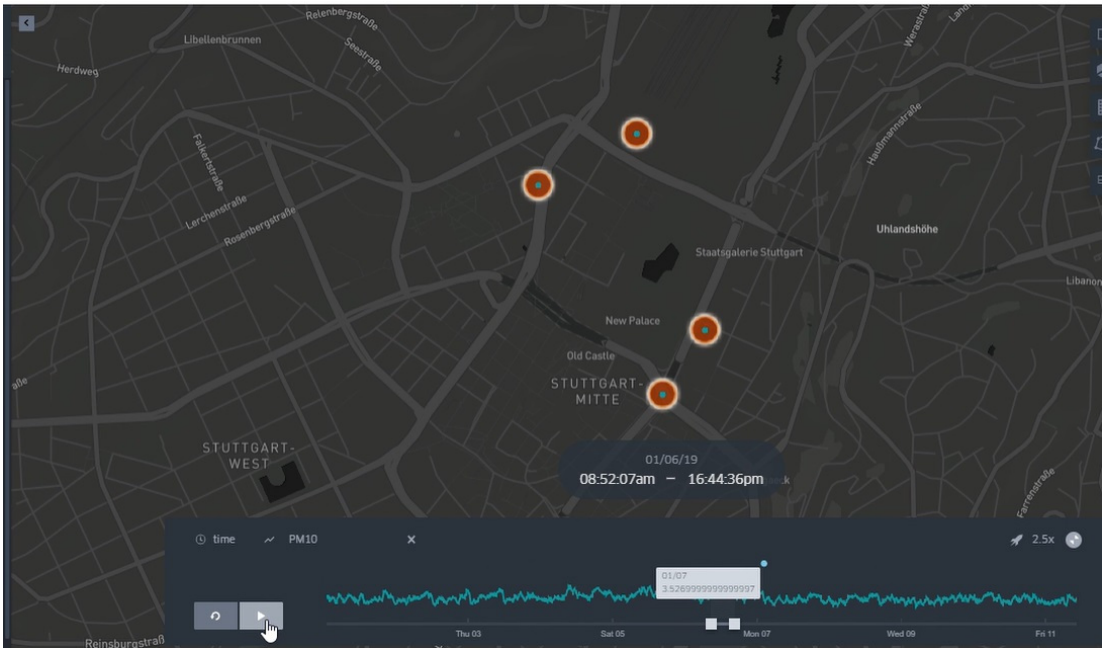


Figure 5.9: PM_{10} concentrations measured by the sensors on map with web interface.

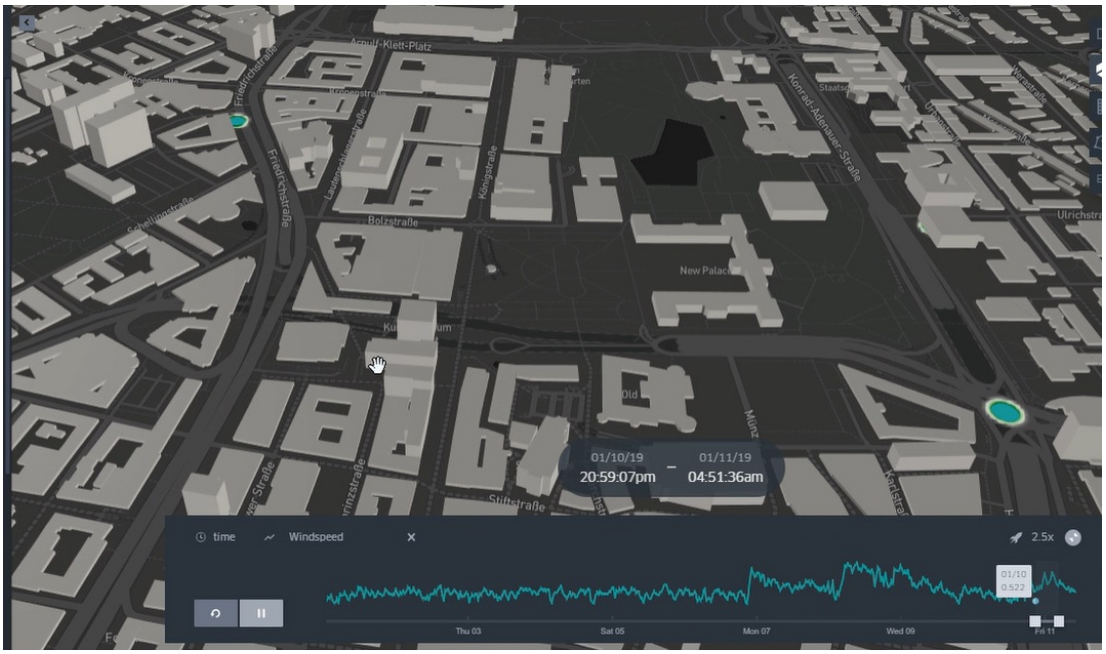


Figure 5.10: Wind speed measured by the sensors on map with web interface having 3D surrounding information.

Thus, HDBSCAN is one of the better clustering option with these advantages, and it is applied on the temporal measurements of sensors and produces interactive filtering output. The hierarchical clustering generates a distance matrix that assists in discovering the clusters similarities and hierarchically merging the similar clusters until the required number of clusters are formed (McInnes and Healy, 2017). This is achieved by minimising the within cluster variance using the error sum of squares as an objective function. Thus, for this work three clusters (classes) ($i.e.$, three: Low, Mild and High) are taken and, starting from respective cluster formation the error sum of squares is kept minimised while deciding their similarities together (Paul and Murphy, 2009).

5.2.3 Transformers Network

In order to provide more detailed comparison and trends analysis, each sensor's nature monitoring using ML approach called Transformers Network predictor model was designed (Wu et al., 2020). The model takes successive time values in terms of parameters as input with sensor's locations and predicts the future dominant (high measurements) value and location with time as the output. The Transformers Network can be represented as an encoder and decoder architecture. This comprises some encoding layers set that process iteratively the input data through each layer one after another in order to generate valuable encodings, followed by a decoder that combines some decoding layers that take the output of the encoder and process further the intermediate output iteratively using their comprehended contextual information to generate an output sequence (Vaswani et al., 2017). The decoding layers also have an additional attention mechanism that draws information from the previous decoding layer output before the following decoding layer draws data from the encodings. Moreover, the feed forward Neural Networks (NN) are used between the sets of encoding and decoding layers in the architecture to perform normalisation steps, and additional outcome processing (Parmar et al., 2018).

The used dataset comprises wind direction, humidity, wind speed, $PM_{2.5}$ and PM_{10}, with temporal resolution $epoch$ and $epoch_j$ ($j \rightarrow 1$ to n) denotes wind direction, humidity, wind speed, $PM_{2.5}$ and PM_{10} at the time j, here 1 represents the first value and n the last value in the dataset. The developed algorithms are trained and tested over the multiple samples that are constructed using the dataset. Moreover, an input with a corresponding three output classes forms a feature vector of a sample. $Window_b$ (a scalar) consecutive values of humidity, wind direction, wind speed, $PM_{2.5}$ and PM_{10} from $epoch_j$ to $epoch_{j+Window_b}$ a feature vector of dimension $Window_b \times 1$ is

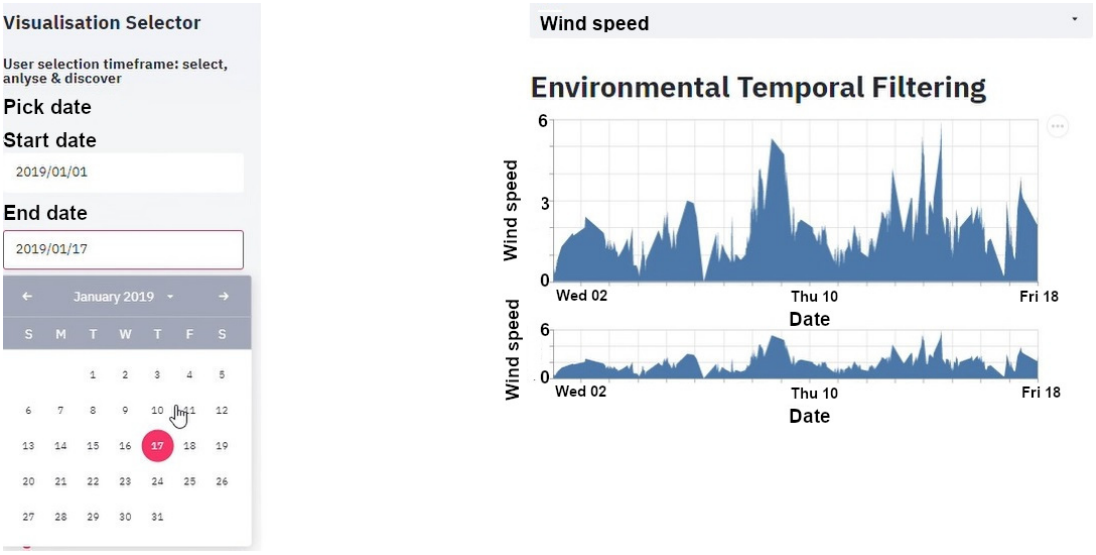


Figure 5.11: Wind speed (WS) data overview in the selected temporal frame visualisation.

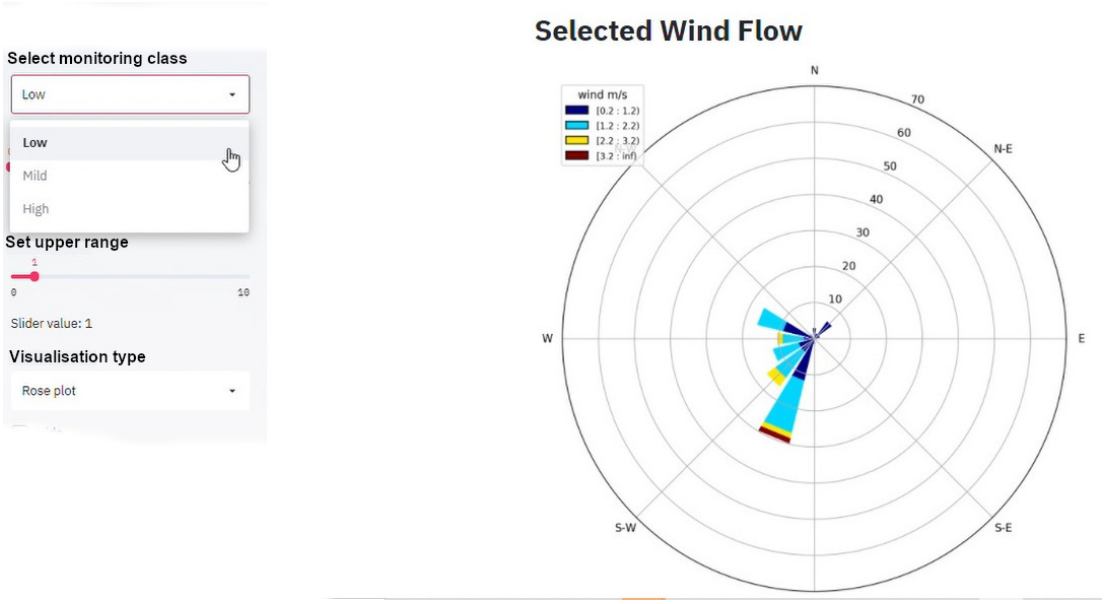


Figure 5.12: Rose plot highlighting the output of clustering sections interactive visualisation.

formed which is the input of the sample for each parameter. $Window_f$ is a scalar which takes considered five parameters successive values after the last value in the input $i.e.$, $epoch_{j+Window_b}$ to $epoch_{j+Window_f}$, and is used to define the sample's output class. Standard deviation (σ) and mean (μ), of humidity, wind direction, wind speed, $PM_{2.5}$ and PM_{10} of the entire dataset are calculated. The multiple class boundaries are constructed using the σ and μ as denoted in Table 5.1. Among $Window_f$, for each class in Table 5.1 the count of values is noted, and the dominant class $i.e.$, the class having the maximum count, is assigned to the sample. Similarly, different samples based on each of the parameters are created by taking $Window_b$ values in the corresponding input from $epoch_j$ to $epoch_{j+Window_b}$ by varying j from 1 to n - $Window_f$, at an increment of 1. The outputs of these samples are designed as discussed above. Therefore, for $Window_b$ values in the input at this stage from $epoch_j$ to $epoch_{j+Window_b}$, there would be five samples sets, based on humidity, wind direction, wind speed, $PM_{2.5}$ and PM_{10}. Here in this analysis the size of $Window_b$ and $Window_f$ are kept equal with user option to predict the next 6 hours. These conditions ensured comprehensive and accurate analysis of the data with respect to independent and different parameter selections.

5.2.4 Visualisation Platform

The developed sensor nature monitoring platform provides clarity of the meteorological and pollution parameters trends, along with spatial visuality in the user selected time frame. This platform is called as Environmental Sensors Visual Prediction Assessment (ESVPA) for sensors nature monitoring. ESVPA also provides tooltiping, brushing and linking for maintaining the transparency and combining different visualisation methods between user-computer efficient interactions (Shneiderman, 1996; Horvitz, 1999). Figure 5.8 provides an overview of ESVPA workflow, along with highlighting the system-user interfaces of visual sensors prediction and analyses. The $System$ combines historical meteorological and pollution parameters temporal database, unsupervised clustering outputs, sensor nature monitoring Transformers Network, structure of various graphs and charts, and accepts user queries. The $User$ raise queries, selects, inspects and views the states of the parameters interactively.

ESVPA uses a time series dynamic stack chart with a calendar selection to visually interact with the parameters as well as the attached spatial information with the help of the interactive map. This is accompanied with a map to provide a detailed time series data inspection of parameters magnitudes with line chart, calendar chart and heat map view option in order to compare the trends among sensors' parameters based on the

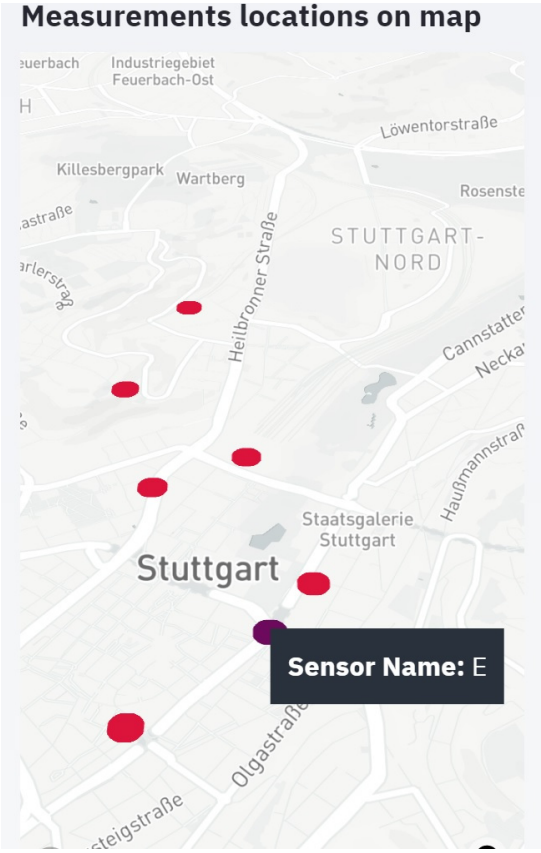


Figure 5.13: Displaying senors measured locations on map.

Figure 5.14: Sensor nature monitoring example.

months during a year, week, and day wise. Figure 5.9, and Figure 5.10 provide a glimpse of the designed web interface where the selected sensors are visualised with spatial and non-spatial information over the map in the specified temporal frame. Over the desire time frame, the user can select the parameters interactively and compare the patterns (as shown in Figure 5.10, Figure 5.11, and Figure 5.12). The interactive platform provides the clustering output visualisation along with sensors nature monitoring. Figure 5.11, and Figure 5.12 show the data overview and clustering results visualisation of designed ESVPA, respectively. Here wind speed and direction parameters are selected to analyse with the help of a stacked chart and rose plots in a chosen time frame. The output from sensors nature monitoring using Transformers Network predictions are represented using Two-Dimensional (2D) map, bar chart, and heat map as shown in Figure 5.14. This work delivers a comprehensive understanding of spatio-temporal sensors data and their measurements. Also, it helps in exploring the relationship between the humidity, wind direction and wind speed $i.e.$, meteorological parameters and pollution parameters like $PM_{2.5}$ and PM_{10}.

5.2.5 Data Used

The temporal datasets of meteorological and pollution parameters are used and analysed in this study. The luftdaten selber messen (Luftdata-se-Stuttgart, 2020) provides city sensors measurements at several locations in Stuttgart, Germany. Moreover, the historical data from 2016 to 2020 from Hauptstaetter Strasse 70173 Stuttgart corner station sensor is also considered (Stadtklima-Stuttgart, 2021). These datasets contain total eleven city centre sensors locations with wind (speed and directions), humidity along with $PM_{10\text{-}2.5}$, measured in a 30-minute time interval (Figure 5.13 shows selected sensors on map). The considered area sensors dataset were organised separately into individual years for each parameter with spatial information attached, using the temporal information with data from the past first, followed by the recent data. This helps to analyse the spatio-temporal trends of meteorological parameters temporal datasets along with sensor's nature monitoring in depth.

5.2.6 Results

The designed algorithms and platform help to perform in depth study of sensors measurements, and also to estimate their nature monitoring for 6hrs in future. This ESVPA is implemented as web-based application using Altair, D3.js, kepler.gl, Streamlit, Keras library (Chollet, 2017) with Tensor-

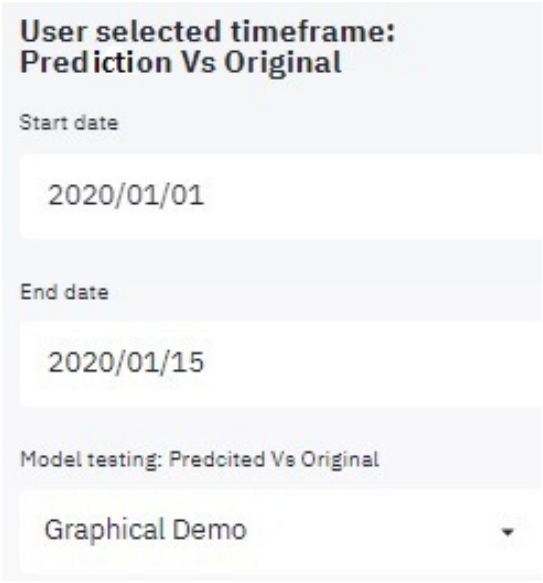


Figure 5.15: Randomly selected date for model validation: Transformers Network visual prediction accuracy analyses, presenting model success-failure (red rows).

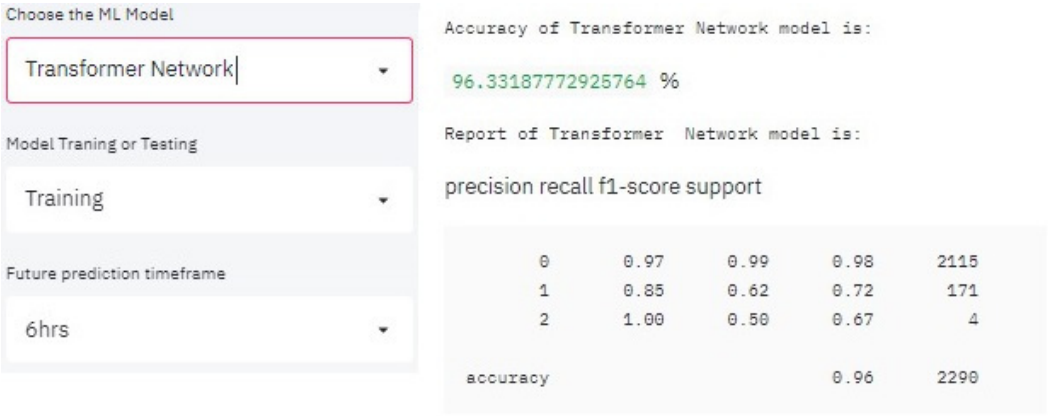


Figure 5.16: (left) Training interactive selection interface, and (right) model accuracy analyses visualisation.

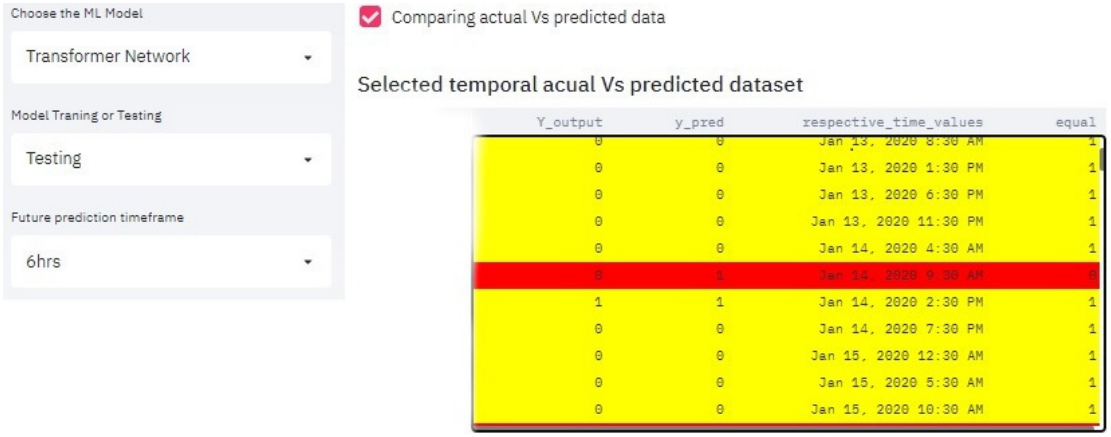


Figure 5.17: (right) Testing interactive selection interface, and (left) comparative network success-failures test visualisation.

Flow in backend in Python and executed on Intel® Core ™ i7- 4770 CPU @3.40 GHz having four cores. The below results and discussion subsections, analysed and validates the outcome of the developed framework. In order to provide a more detailed comparison of the selected parameters, the visualisation of historical measurements along with spatio-temporal information was explored using the available interactive option of data overview in the designed platform. Figure 5.9 and Figure 5.10 show the historical data visualisation for the user selected time frame of PM_{10}, and wind flow on the map with the help of line charts, which help to connect spatio-temporal information with the respective sensors measurements visually. In these figures, the radii of the circles denote the magnitude of the selected parameter with time (more the magnitude larger the radius), and the colour of the circles represents the selected parameter. The platform was also used to visualise the output of the HDBSCAN clustering (as shown in Figure 5.11 and Figure 5.12). The HDBSCAN an unsupervised hierarchical clustering approach, helped towards inferring the trends and inner structure of the meteorological and pollution parameters dynamically. Figure 5.11 shows the clustering outcome and its visualisation on wind speed as a selected parameter over the considered time frame. Similar steps were carried out to perform the analyses on the rest of the other considered parameters. Here the class value ranges of each assigned class were also displayed and compared. Moreover, the performed clustering with visualisation helped the user unboxed the complexities of datasets and their available trends in the best possible way.

Furthermore, the obtained wind rose plot helped visualise wind speed and direction in a circular format in the same graph. The length of each

spoke around the circle indicates the number of times (count) that the wind blows from the indicated direction. Colors along the spokes indicate classes of wind speed. Figure 5.12 shows the generated wind rose plot on the selected temporal frame. Besides, each different color denote the wind speed divided into value range boundaries at the differences (within the class assigned maximum and minimum value) with varying spoke length and direction highlighting the wind blows count from the indicated directions in this study. Figure 5.14 shows the output of sensors nature monitoring using Transformers Network predictions. These options were integrated together with the selection of the desired user query. Here, the map highlights the location of the respective selected sensor, with day wise sensors' measurements visualised with the bar chart. The heat map and bar chart are linked together with the map selection in the visualisation interface. Moreover, the linked heat map represents the magnitude of the selected parameter values on 30 minute time resolution ($i.e.$, query the data after every 30 minute values) which is denoted by the color density ranging from green (lower value) to red (higher value). The selected parameter can be any out of $i.e.$, $PM_{2.5}$, wind speed, PM_{10} and humidity, where higher values (range) over time had been assigned an intense hue tone of the respective color (mentioned as above) in the heat map. The attached bar chart represents the magnitude of the selected parameter values on days in the week time resolution (query the data in each day values in a week). In Figure 5.14 PM_{10} is considered as a parameter to predict the sensor nature with map, bar chart and heat map visualisation. Furthermore, the available option of the interactive time frame selected by the user including desire parameters selection, would help the user to compare and visualise the trends in a more detailed manner.

Figure 5.16 and Figure 5.17 show the designed network's achieved accuracy with the selection of the desired user query. Here, precision and recall values for predicting dominant speed for various classes for January month (of Stuttgart) are represented in Figure 5.16, with values reaching above 61% for all the classes and achieved total accuracy was 96.33%. Figure 5.15 shows the randomly selected date for model validation and highlights the obtained Transformers Network visual prediction accuracy analyses with the models' success-failure (red table's rows). Furthermore, this has supported sensitivity analyses for calculating the success and failure of the model highlighted with color and dynamic interaction. Here network's success was represented by yellow color and failure with red. These color combinations were used to deliver more insights thereby making the understanding for the user more straightforward and unboxing the complexities of ML. Thus, this platform (all together) helped to discover all the possible changes by enhancing the ability to dig in detail insight of the

data with accuracy for each of the considered meteorological and pollution parameters as per the user choice visually.

5.2.7 Discussion

The meteorological and pollution parameters hierarchical clustering highlighted the trends for a selected parameter which is analysed in the clustering diagram, with each class assigned lower and upper value ranges. HDBSCAN performed exploratory data analysis as it is a fast and robust algorithm that helped to work over the unsmoothed temporal meteorological and pollution parameters to return meaningful clusters. For the rose plot colored scale map, a sequential scale color brewer was used to represent the classes (low, mild and high) with the color frequency differentiating the class of low value range from the class of high values. Moreover, using the multi hue progression of blending supported by choosing from the least to the most opaque shades concerning value ranges occurring in the clusters represent low to high values. The 2D map view of all the selected sensors on the map, along with time based data filtering query with tool tipping helps to easily interact and visualise all the information together in one platform as shown in Figure 5.13. The values in the dataset of each year for the selected parameter over the considered time frame that joined together earlier in the clustering are more similar than those joined together later. The within total cluster variance is minimised during clustering. The paired clusters with a minimum in between cluster distance at each step are merged. Therefore, in the result it is observed that in February, a higher magnitude of wind flow occurs over 2016 to 2020.

On the other hand, the Transformers Network helped to estimate sensors nature interactively. The input sample consisted of $Window_f$ consecutive values from the data with five features of $PM_{2.5}$, PM_{10}, humidity, and wind (speed and direction) providing temporal information and Transformers Network operations are able to detect trends and features. During the sample designing phase, the output classes were decided statistically using μ and σ of the total values particular to a year's data set of respective parameter ($i.e.$, anyone out of five), thereby representing the dataset better. Moreover, the total samples for a given year were divided into training and testing samples with a ratio of $7:3$ ($i.e.$, 70% of the total samples for training and rest for testing). The dynamic network metrics analyses (total accuracy, precision and recall) of the Transformers Network supported with interactive visualisation with several options help the user verify and understand these metrics for the selected parameter. Visual exploration has also been incorporated to make ML more easily understandable and explainable in the sense that network insights can be explainable. Moreover, the

developed ESVPA for sensors nature monitoring is utilised to provide the interactive selections for the considered environmental data for temporarily analysing the concurred pattern in the dataset. ESVPA is also compared with existing literature that are near to the developed framework. AQTA (see subsection 5.1.2), has provided visual analyses platform of air quality data with time but lacks sensors nature monitoring. It discovers temporal relationships among complex air quality data, on a small scale for each sensor (individually) while missing the spatial information. However, the developed ESVPA connects temporal, spatial and non-spatial information together visually. Further, the time series analyses were enhanced using the unsupervised HDBSCAN clustering on a series of (above mentioned) parameters. Therefore, ML approach based on Transformers Network is integrated with the in depth sensors nature understanding and trends, that take successive time values of parameters as input with sensors' locations and predict the future dominant (highly measured) values with the location in time as the output. This makes ESVPA a work extension that provides a big picture of sensors' nature monitoring and temporal data measurement analyses. This helped in making the data trends analyses and sensors nature monitoring accessible and comparable in the time domain with user involvement.

5.2.8 Conclusion

In this work, ESVPA, an interactive web visualisation is successfully designed and demonstrated for time series meteorological and pollution parameters. The temporal datasets are analysed using the unsupervised HDBSCAN clustering on a series of these parameters. Furthermore, for sensors nature understanding and trends, Machine Learning (ML) approach called the Transformers Network predictor is also integrated, which takes successive time values of parameters as input with sensors' locations and predicts the future dominant (highly measured) values with location in time as the output. The interactive platform for meteorological and pollution parameters would help to plan the future with more renewable resources awareness and understanding. The designed visualisation platform (a small demonstration version) in this work could be further improved with the ensemble of advanced visualisation approaches. The selected environmental data variations are compared and analysed in the spatio-temporal frame to provide detailed estimates on change in the average conditions in a region over the years.

5.3 Chapter Summary

This chapter presents the machine learning models visual assessments to provide the user with the clarity of what machine learning is doing, integrating successfully on real world temporal datasets (meteorological and pollution parameters) with the designed model failure and success analyses interactively.

AQTA is an interactive system to support visual analyses of air quality data with time. This interactive AQTA allows the seamless integration of predictive models and detailed patterns analyses. The initial findings are presented that corroborate the city's COVID lockdown (the year 2020) conditions and sudden changes in patterns, highlighting the improvements in the pollutants concentrations. AQTA, thus, successfully discovers temporal relationships among complex air quality data, interactively in different time frames, by harnessing the user's knowledge of factors influencing the past, present and future behaviour, with the aid of ML models. Further, this study also reveals that the decrease in the concentration of one pollutant does not ensure that the surrounding air quality would improve as other factors are interrelated. The second approach, ESVPA is an advancement of adding more sensors and visually analysing them all together. The time series are analysed using the unsupervised HDBSCAN clustering on a series of (above mentioned) parameters. Furthermore, ML approach based on Transformers Network is integrated with the in depth sensors nature understanding and trends, that takes successive time values of parameters as input with sensors' locations and predicts the future dominant (highly measured) values with the location in time as the output. The selected parameters variations are compared and analysed in the spatio-temporal frame to provide detailed estimations on how average conditions would change in a region over time. This work would help to get a better insight into the spatio-temporal data. Moreover, based on the explanations of XAI (section 2.4), this work has developed the following XAI techniques:

1. The user could choose among different ML models by analysing the performance of each selected model (MCNN, LSTM, RF, K-NN and SVM) in terms of total accuracy and difference metrics incorporated with the interactive display through various graphs and charts such as square-time charts with tooltip, temporal circle mark chart coupled with a histogram.

2. The user could change the future time frame allowing for a different prediction duration, and compare the results of different time frames and the outcome of each model to decide which prediction

algorithms are better and provides sufficient information to make a decision.

3. The respective parameters (meteorological and pollution parameters) can be selected and compared where both predicted and actual results are visualised in the interactive graphs.

4. The clusters formed in the temporal environmental data can be learnt by ML algorithms and then predict the future trends with respective sensor location.

6

Conclusion & Future Work

The beginning of the thesis (section 1.5) presented the overall research objectives. The research aimed to develop ML interpretability methods for the environmental data that accommodates a robust web framework comprising ML architectures for time series prediction and interactive visualisation methods using VA concepts wrapped over ML models. This developed interactive visualisation system for environmental data assimilates ML architectures visual prediction, sensors' spatial locations, measurements of the parameters, detailed pattern analyses, and change in conditions over time. The experiments were conducted using various meteorological and pollution datasets to ascertain the performance of the developed system. The subsequent paragraphs highlight the contributions of multiple methods developed in this research.

6.1 Conclusions

Chapter 3 developed six ML algorithms viz, 1DLSTM, 1DRF, 1DSVM, 1DS, 1DM, and MCLT, for predicting and analysing the environmental data. The contributions of these algorithms were:

1. they have not applied any smoothening and noise removal techniques and are based on a classification approach,

2. more number of classes are integrated into the architectures for in depth analyses,

3. 1DS and 1DM are enhanced versions of 1DLSTM, 1DRF, 1DSVM in terms of incorporating multiple features and more number of classes,

4. 1DS and 1DM architectures have been missing the goodness of memory units to retain the features learnt by neurons from the previous training iterations and hence advanced by MCLT architecture,

5. multiple features were manually designed in the input layers of these ML algorithms, whereas the intermediate layers in developed deep learning architectures (1DS, 1DM, MCLT) learnt their respective features automatically during training,

6. MCLT architecture was advanced with the goodness of both multiple CNN and LSTM,

7. presence of densely connected convolutional layers helped to learn features of other convolutional layers as well, and

8. the objective behind using more number of classes with a close difference range helped to identify more details, and results behaved very close to regression with the best accuracy,

The above aspects provided more information and enhanced the performance of the techniques. The experiments with meteorological and pollution parameters, as well as their findings, showed good accuracies. At the time of developing these methods, using LSTM, CNN, MCLT (ensemble architecture) integrated with multiple designed features for classification of environmental datasets, the methods were unique in prediction literature for environmental data. However, in the above discussed ML based methods for prediction, there was a lack of visualisation as required in VA. Thus, Chapter 4 and Chapter 5 were necessary that helped in the visualisation of different patterns in the dataset for different time frames interactively.

Chapter 4 emphasised on the seasonality deduction for the pollution parameters in relationship with the meteorological parameters. The contributions of these techniques were:

1. provide interactive selections of considered meteorological and pollution parameters to analyse the concurred temporal patterns in the dataset, for each quarter (Q_1, Q_2, Q_3, and Q_4) over the years,

2. hierarchical cluster analyses to highlight the trends of any given pair of quarters (over the years),

3. each quarter was further analysed using the initial 15 days and last 15 days that helped in making the seasonality analyses tests easy, user interactive and comparable in the same quarter,

Q1	Q2	Q3	Q4
From 1st to 15th, and 23th to 27th in years 2015 to 2017 highest with constantly increasing level, while little controlled and reduced in 2018 and 2019.	13th to 15th increased in 2015 to 2016 and reached highest during years 2018 to 2019. From 27th to 30th High in 2015 to 2016 and then reduced to lowest in 2019	Frequently changing PM10 concentration from lowest to increasing in 27th to 30th with the concentration reached highest during 2015 to 2019.	From 2nd to 4th, and 13th to 15th in 2015 to 2019 PM10 concentration measured highest.

Figure 6.1: The clustering output for PM_{10} for all seasonal quarters Q_1 - Q_4 over 2015 to 2019.

4. used a sequential scale of color brewer blues scale color map to show several classes with the color frequency differentiating low values class from high values class.

5. findings such as:

   a) NO and NO_2 concentrations were high in Q_3 autumn, and Q_4 winter over 2015 to 2019 respectively. Both are strongly correlated to each other with similar trends over the years,

   b) Figure 6.1 summarises the clustering output for PM_{10} for all seasonal quarters Q_1 - Q_4 over 2015 to 2019. Similar analyses exist for other parameters too.

6. provided foreknowledge of meteorological parameters nature in relation to pollution parameters of an area.

At the time of proposing these seasonality deductions, the contributions were unique in terms of no available detailed analyses for cities meteorological and pollution parameters together. Moreover, an improved approach was required that combines more environmental data, correlation analysis, a temporal heat map and a better interactive visualisation integrating with the above developed ML architectures visual predictors for multiple parameters in depth analysis for various time frames in a robust web platform.

Chapter 5 provided an enhanced visualisation platform, integrated with time series data visual predictive assessments and sensors nature analyses. AQTA platform deliverables were:

1. it allowed the seamless integration of predictive models and detailed patterns analyses visualisation,

2. back-and-forth dialogue with the designed multiple ML models and comparisons for better visual predictive assessments in different time conditions for chosen parameters,

3. it provided data selection, display, visualisation of past, present, future and correlation structure among air parameters through various interactive charts, highlighting the predictive models effectiveness,

4. it was revealed (supported with detailed analyses) that the decrease in the concentration of one pollutant does not ensure that the surrounding air quality has been improved as other factors are interrelated,

5. focused on integrating and linking the simple charts representation to discover complex air quality parameters interactively in various time frames, with options to have,

   a) a visual data overview (history and present),

   b) future prediction along with model success, failure comparison, and

   c) a correlation structure of their interrelationships.

6. analysis showed that during summer and autumn of the year 2019, PM_{10} trends were alike as in the previous years 2017 to 2018 with a few reductions. Furthermore, there is depletion in PM_{10} concentrations during the summer and autumn of the year 2020 probably due to the strict lockdown and movements restrictions,

7. it was observed that the local wind could often develop that does not cause high magnitude winds, but plays an essential role in local ventilation of the city areas and determines the spread of air pollutants,

AQTA provided an add-on to the existing literature in terms of air quality multiple time series datasets, dynamic visual predictions along with its detailed analyses, comparisons and validation with reality. The AQTA used for in depth analyses of one sensor (or any sensor analyses) was further advanced by including the information about different sensors locations and correlation with each other in ESVPA. The ESVPA is supported with an add-on to the sensor nature's monitoring, which motivated the geospatial

data visual assessments web interface. Following were the contributions of the ESVPA:

1. ML Transformers network was trained for modelling the future dominant (high measurements) sensor locations with time as the output.

2. the Transformers network analyses using total accuracy, precision and recall supported with interactive visualisation helped the user understand the outcomes. The user can also interact with training and testing phases of the network modelling.

3. the platform provided comparison and analyses in the spatio-temporal frame along with detailed estimates on changes in meteorological and pollution parameters.

4. these techniques also form a part of XAI.

Varieties of environmental datasets having temporal values for more than 30 years, have been used for conducting several experiments to evaluate the effectiveness of the developed techniques, and the main findings have been given above. The results obtained from these experiments show that the developed techniques are able to discover the temporal relationships among complex environmental data interactively in different time frames. Furthermore, with the present combinations of neurons and features maps, the accuracies achieved by the ML architectures have been significant. ML frameworks have shown their potential to resolve highly volatile environmental data with detailed analyses, and understanding. The achieved accuracies can be further improved with more advanced ML architectures, more variety of environmental datasets, a higher number of feature maps, and neurons. Moreover, these require better hardware resources and system support.

A mixture of both unsupervised and supervised clustering techniques have been used for environmental data. This work has also provided and demonstrated a comprehensive outlook of the possible analyses that could be conducted on meteorological and pollution parameters utilising the advantages of these two domains in parallel intelligently. Many cities are providing open environmental data, but the online analysis capabilities in their open data platforms are usually weak or non-existent. Moreover, the following motive of integrating the ML and VA together to enhance the online analyses capabilities of the open environmental data have also been accomplished in these designed techniques. This research work also devises the initiative to fill this gap for missing detailed online environmental

data analyses capabilities. The designed framework is implemented for Stuttgart and Netherlands sensors locations. However, it can be applied to any number of sensors for any given location (area) with some ML tuning and training of the respective datasets.

6.2 Future Work

The future work would involve improving the current techniques for better web based visualisation and implementation in different environmental application areas. Integrating with more advanced VA techniques, focusing more on the visual exploration and interactive visualisation components with end-user analysis and feedback supported in the loop interactively, would make the techniques more user friendly and more understandable. City planner and experts involvement and suggestions in order to advance this work as a real time product implementation and enhancement would be highly valuable for smartly planning of the cities.

Furthermore, deep learning architectures perform better when a variety of training samples are used. This would require integration of more sensor data. The impact of data accuracy, time frame, volatility of meteorological and pollution parameters further needs to be explored using these approaches, as it would give a better idea of the usefulness of the methods with environmental sensors data. With the use of better hardware resources like GPUs, deep networks consisting of more convolutional and fully connected layers along with higher number of feature maps and neurons could be implemented for higher environmental data predictive accuracies. Although the time taken in training and testing the algorithms is hardware dependent, still for the same hardware, the time taken by the current approaches could be compared with other approaches. Moreover, the automation potential of the methods for real time purposes can be studied. The algorithms developed in this thesis can be used for the real time inference of the spatio-temporal environmental data, enabling the interactive exploration processes. Scalability in terms of larger number of spatial sensors, storage and fast retrieval and display of the data, can be explored. The integration of high-end web services for supporting smooth and fast processing of both frontend and the backend would help the user make quick decisions and further devising detailed environmental data analyses for respective problem solutions. Also, an Augmented Reality (AR) mobile supported Application (App) for the above developed techniques would conveniently support the analyses. This work could be extended by making a sort of grid covering the territory under study, and constructing a sort of network of neurons influencing each other at the vicinity.

Author's Work

1. Harbola, S. and Coors, V. (2018), 'Geo-Visualisation and Visual Analytics for Smart Cities: A Survey', *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-4/W11, https://doi.org/10.5194/isprs-archives-XLII-4-W11-11-2018, 11–18

2. Harbola, S. and Coors, V. (2019a), 'One Dimensional Convolutional Neural Network Architectures for Wind Prediction', *Energy Conversion and Management*, 195, https://doi.org/10.1016/j.enconman.2019.05.007, 70–75

3. Harbola, S. and Coors, V. (2019b), 'Comparative analysis of LSTM, RF and SVM Architectures for Predicting Wind Nature for smart city planning', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-4/W9, https://doi.org/10.5194/isprs-annals-IV-4-W9-65-2019, 65–70

4. Harbola, S. and Coors, V. (2019c), 'Convolutional Neural Network Architectures for Wind Analysis,', *EAWE PhD Seminar 2019 29-31 Oct 2019 Nantes, France*, https://eawephd2019.sciencesconf.org/285035,

5. Harbola, S. and Coors, V. (2020), 'Seasonality Deduction Platform : For PM_{10}, $PM_{2.5}$, NO, NO_2 and O_3 in Relationship with Wind Speed and Humidity', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, VI-4/W2, https://doi.org/10.5194/isprs-annals-VI-4-W2-2020-71-2020, 71–78

6. Harbola, S., Koch, S., Ertl, T., and Coors, V. (2021a), 'Air Quality Temporal Analyser: Interactive temporal analyses with visual predictive assessments', *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, https://doi.org/10.2312/envirvis.20211083

7. Harbola, S. and Coors, V. (2021a), 'Deep learning model for wind forecasting', *PFG – Journal of Photogrammetry, Remote Sensing*

and Geoinformation Science, 10, https://doi.org/10.1007/s41064-021-00185-6

8. Harbola, S. and Coors, V. (2021b), 'An Interactive Platform For Environmental Sensors Data Analyses', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, VIII-4/W1-2021, https://doi.org/10.5194/isprs-annals-VIII-4-W1-2021-57-2021, 57–64

9. Harbola, S., Storz, M., and Coors, V. (2021b), *Augment Reality for Windy-cities:3D Visualisation of future wind nature analysis in city planning* (Springer, (To appear, accepted on 2020- July -20))

10. Brennenstuhl, M., Gruen, M., Harbola, S., Koukofikis, A., Padsala, R., Schaaf, M., Coors, V., and Voss, U. (2021), 'CFD Simulation and visualization based investigation of small wind turbine potential: A case study "Neuer Stöckach" for Stuttgart', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, VIII-4/W1-2021, https://doi.org/10.5194/isprs-annals-VIII-4-W1-2021-17-2021, 17–24

All above links were last followed on 2021-Sept-5

Bibliography

Ahvenniemi, H., Huovila, H., Pinto-Seppä, I., and Airaksinen, M. (2017), 'What are the differences between sustainable and smart cities?', *Cities*, 60, https://www.sciencedirect.com/science/article/pii/S0264275116302578, 234–245. (Cited on page 19.)

Aigner, W. (2013), 'Interactive visualization and data analysis: visual analytics with a focus on time.', *Habilitation dissertation, subject Practical Computer Science submitted at the Technical University of Vienna*, http://publik.tuwien.ac.at/files/PubDat_227076.pdf. (Cited on page 121.)

Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011), 'Visualization of Time-Oriented Data', *Springer-Verlag London Limited*, https://doi.org/10.1007/978-0-85729-079-3. (Cited on page 35.)

Aissou, S., Rekioua, D., Mezzai, N., Rekioua, T., and Bacha, S. (2015), 'Modeling and control of hybrid photovoltaic wind power system with battery storage', *Energy Conversion and Management*, 89, https://doi.org/10.1016/j.enconman.2014.10.034, 615–625. (Cited on page 33.)

Albino, V., Berardi, U., and Dangelico, R. (2015), 'Smart Cities: Definitions, Dimensions, Performance, and Initiatives', *Journal of Urban Technology*, 22, https://doi.org/10.1080/10630732.2014.942092, 2015. (Cited on page 18.)

Andrade, A. (2019), 'Best Practices for Convolutional Neural Networks Applied to Object Recognition in Images', *ArXiv*, https://arxiv.org/abs/1910.13029. (Cited on pages 24 and 27.)

Andrienko, G., Andrienko, N., Keim, D., MacEachren, A. M., and Wrobel, S. (2011), 'Challenging problems of geospatial visual analytics', *Journal of Visual Languages & Computing*, 22/4, https://www.sciencedirect.com/science/article/pii/S1045926X11000280, 251–256. (Cited on page 13.)

Andrienko, N. and Andrienko, G. (2003), 'Coordinated views for informed spatial decision making', *In: Proceedings International Conference on Coordinated and Multiple Views in Exploratory Visualization - CMV 2003*, https://doi.org/10.1109/CMV.2003.1215002. (Cited on pages 36 and 105.)

Andrienko, N. and Andrienko, G. (2013), 'Visual analytics of movement: An overview of methods, tools and procedures', *Information Visualization*, 12/1, https://doi.org/10.1177/1473871612457601, 3–24. (Cited on page 20.)

Arentze, T. and Timmermans, H. (2000), 'A spatial decision support system for retail plan generation and impact assessment', *Transportation Research Part C: Emerging Technologies*, 8/1, https://www.sciencedirect.com/science/article/pii/S0968090X00000103, 361–380. (Cited on page 14.)

Arthur, T. D. and Owen, M. D. (2003), 'Temporal, spatial and meteorological variations in hourly Pm2.5 concentration extremes in New York City', *Atmospheric Environment*, 38, https://doi.org/10.1016/J.ATMOSENV.2003.12.020, 1547–1558. (Cited on page 84.)

Badam, K., Zhao, J., Sen, S., Elmqvist, N., and David, E. (2016), 'TimeFork: Interactive Prediction of Time Series', *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 7/12, https://doi.org/10.1145/2858036.2858150, 5409–5420. (Cited on pages 36, 104, and 105.)

Bayulken, B. and Huisingh, D. (2015), 'Are lessons from eco-towns helping planners make more effective progress in transforming cities into sustainable urban systems: A literature review (part 2 of 2)', *Journal of Cleaner Production*, 109, https://doi.org/10.1016/j.jclepro.2014.12.099. (Cited on page 20.)

Bhattacharya, D. and Painho, M. (2017), 'Smart Cities Intelligence System (SMACiSYS) Integrating Sensor Web with Spatial Data Infrastructures (sensdi)', *Journal of the Knowledge Economy*, 4/4, http://hdl.handle.net/10362/28046, 21–28. (Cited on pages 13 and 17.)

Biber, E. (2013), 'The challenge of collecting and using environmental monitoring data', *Ecology and Society*, 18/4, http://dx.doi.org/10.5751/ES-06117-180468, 68. (Cited on page 1.)

Bogue, R. (2008), ‘Environmental sensing: strategies, technologies and applications’, *Earth-Science Reviews*, 28, https://doi.org/10.1108/02602280810902550, 275–282. (Cited on pages 37 and 106.)

Bollen, J., Mao, H., and Zeng, X. (2011), ‘Twitter mood predicts the stock market’, *Journal of Computational Science*, https://doi.org/10.1016/j.jocs.2010.12.007, 135–158. (Cited on pages 36 and 105.)

Brennenstuhl, M., Gruen, M., Harbola, S., Koukofikis, A., Padsala, R., Schaaf, M., Coors, V., and Voss, U. (2021), ‘CFD Simulation and visualization based investigation of small wind turbine potential: A case study "Neuer Stöckach" for Stuttgart’, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, VIII-4/W1-2021, https://doi.org/10.5194/isprs-annals-VIII-4-W1-2021-17-2021, 17–24. (Not cited.)

Bröring, A., Vial, D., and Reitz, T. (2014), ‘Processing Real-Time Sensor Data Streams for 3D Web Visualization’, *Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming*, 14/9, https://doi.org/10.1145/2676552.2676556„ 72–80. (Cited on page 14.)

Campello, J. G. B., Moulavi, D., and Sander, J. (2013), ‘Density-Based Clustering Based on Hierarchical Density Estimates’, *Pacific-Asia Conference on Knowledge Discovery and Data Mining Proceedings of the Springer*, https://link.springer.com/chapter/10.1007/978-3-642-37456-2_14. (Cited on page 123.)

Cao, N. and Cui, W. (2016), ‘Introduction to text visualization’, *Springer Nature*, 1/12, https://doi.org/10.2991/978-94-6239-186-4,. (Cited on page 14.)

Card, S. K., Mackinlay, J. D., and Shneiderman, B. (1999), ‘Readings in Information Visualization: Using Vision to Think’, *Information Visualization - IVS*, https://dl.acm.org/doi/10.5555/300679. (Cited on page 12.)

Chan, P. and Mohoney, M. (2005), ‘Semantic interaction for visual text analytics’, *proceedings of the IEEE conference on data mining*, 123, https://doi.org/10.1145/2207676.2207741, 52–55. (Cited on pages 36 and 105.)

Chen, C. and Zhao, B. (2011), ‘Review of relationship between indoor and outdoorparticles: I/O ratio, infiltration factor and penetration factor’, *Atmospheric Envi-ronment*, https://doi.org/10.1016/j.atmosenv.2010.09.048, 1–9. (Cited on pages 84 and 120.)

Chen, M., Mao, S., and Liu, Y. (2014), 'Big Data: A Survey', *Mobile Networks and Applications volume*, 19/4, https://link.springer.com/article/10.1007/s11036-013-0489-0, 171–209. (Cited on pages 13 and 14.)

Cho, R. (2018), 'Artificial Intelligence—A Game Changer for Climate Change and the Environment', *State of the Planet, News from the Columbia Climate School*, https://news.climate.columbia.edu/2018/06/05/artificial-intelligence-climate-environment/. (Cited on page 2.)

Chollet, F. (2017), 'Deep Learning with Python', *Manning Publishing Platform*, https://www.oreilly.com/library/view/deep-learning-with/9781617294433/. (Cited on pages 56, 70, 108, and 129.)

Choo, J. and Liu, S. (2018), 'Visual Analytics for Explainable Deep Learning', *IEEE Computer Graphics and Applications*, 38/4, https://doi.org/10.1109/MCG.2018.042731661, 84–92. (Cited on pages 4, 22, 23, and 82.)

Chudá, D. (2007), 'Visualization in Education of Theoretical Computer Science', *Proceedings of the 2007 International Conference on Computer Systems and Technologies*, 24/6, https://doi.org/10.1145/1330598.1330687. (Cited on page 14.)

Church, R. L. and Cova, T. J. (2000), 'Mapping evacuation risk on transportation networks using a spatial optimization model', *Transportation Research Part C: Emerging Technologies*, 8/1, https://www.sciencedirect.com/science/article/pii/S0968090X0000019X, 321–336. (Cited on page 14.)

Claessen, J. H. and van Wijk, J. J. (2011), 'Flexible Linked Axes for Multivariate Data Visualization', *IEEE Transactions on Visualization and Computer Graphics*, 17/12, https://doi.org/10.1109/TVCG.2011.201„ 2310–2316. (Cited on page 14.)

Clevert, D., Unterthiner, T., and Hochreiter, S. (2016), 'Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)', *Proceedings of the International Conference on Learning Representations*, 27/6, https://arxiv.org/abs/1511.07289, 21–28. (Cited on pages 57 and 71.)

Colak, I., Sagiroglu, S., and Yesilbudak, M. (2012), 'Data mining and wind power prediction: A literature review', *Renewable Energy*, 46/6, https://doi.org/10.1016/j.renene.2012.02.015, 241–247. (Cited on page 33.)

Congalton, R. and Green, K. (2008), 'Assessing the accuracy of remotely sensed data: principles and practices', *CRC press*, 10/2, https://doi.org/10.1201/9781420055139, 200. (Cited on page 31.)

Cormack, R. M. (1971), 'A Review of Classification', *Journal of the Royal Statistical Society*, 134/3, https://doi.org/10.2307/2344237, 321–367. (Cited on page 86.)

Daraeepour, A. and Echeverri, D. P. (2014), 'Day-ahead wind speed prediction by a Neural Network-based model', *PES Innovative Smart Grid Technologies conference, ISGT - 2014. Proceedings of the IEEE*, https://doi.org/10.1109/ISGT.2014.6816441, 220 – 226. (Cited on page 34.)

Dasgupta, A. and Kosara, R. (2011), 'Adaptive Privacy-Preserving Visualization Using Parallel Coordinates', *IEEE Transactions on Visualization and Computer Graphics*, 17/12, https://ieeexplore.ieee.org/document/6064989, 2241–2248. (Cited on page 14.)

Dasgupta, A., Chen, M., and Kosara, R. (2012), 'Conceptualizing Visual Uncertainty in Parallel Coordinates', *Computer Graphics Forum*, 31/3pt2, https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2012.03094.x, 1015–1024. (Cited on page 14.)

De Giorgi, M., Campilongo, S., Ficarella, A., and Congedo, P. (2014), 'Comparison Between Wind Power Prediction Models Based on Wavelet Decomposition with Least-Squares Support Vector Machine (LS-SVM) and Artificial Neural Network (ANN)', *Energies*, 7, https://doi.org/10.3390/en7085251, 5251–5272. (Cited on pages 34 and 51.)

De Giorgi, M. G., Ficarella, A., and Russo, M. G. (2016), 'Short-term wind forecasting using artificial neural networks (ANNs)', *WIT Transactions on Ecology and the Environment*, 121, https://doi.org/10.1109/ICISET.2016.7856485, 197–208. (Cited on page 34.)

Del Fatto, V., Laurini, R., Lopez, K., Loreto, R., Milleret-Raffort, F., Sebillo, M., Sol-Martinez, D., and Vitiello, G. (2007), '. Potentialities of chorems as visual summaries of geographic databases contents', *Advances in Visual Information Systems*, 4781/11, https://link.springer.com/chapter/10.1007/978-3-540-76414-4_52„ 537–548. (Cited on page 15.)

El-Fouly, T. H. and El-Saadany, M. M. E. F.and Salama (2008), 'One day ahead prediction of wind speed and direction', *Energy Conversion, IEEE Transactions on*, 24, https://doi.org/10.1109/TEC.2007.905069, 191–201. (Cited on page 34.)

Elminir, H. K. (2005), 'Dependence of urban air pollutants on meteorology', *Science of Total Environment*, 350, https://doi.org/10.1016/j.scitotenv.2005.01.043, 225–237. (Cited on page 84.)

Endert, A., Fiaux, P., and North, C. (2012), 'Semantic interaction for visual text analytics', *Proceedings of the 2012 CHI Conference on Human Factors in Computing Systems*, 123, https://doi.org/10.1145/2207676.2207741, 52–55. (Cited on pages 37 and 106.)

Filik, U. B. and Filik, T. (2017), 'Wind Speed Prediction Using Artificial Neural Networks Based on Multiple Local Measurements in Eskisehir', *International Conference on Energy and Environment Research, ICEER - 2016*, 107, https://doi.org/10.1016/j.egypro.2016.12.147, 264–269. (Cited on page 34.)

Flores, B. E. (1986), 'A pragmatic view of accuracy measurement in forecasting', *Omega*, 14, https://doi.org/10.1016/0305-0483(86)90013-7. (Cited on pages 61 and 79.)

Fraser, M. P., Yue, Z. W., and Buzcu, B. (2003), 'Source appointment of fine particulate matter in Houston TX, using organic molecular markers', *Atmospheric Chemistry and Physics*, 37, https://doi.org/10.5194/acp-18-15601-2018, 2117–2123. (Cited on page 84.)

Fu, T. C. (2011), 'A review on time series dagta mining', *Engineering Applications of Artificial Intelligence*, https://doi.org/10.1016/j.engappai.2010.09.007. (Cited on pages 36 and 105.)

Fu, Y. and Zhang, X. (2017), 'Planning for Sustainable Cities? A Comparative Content Analysis of the Master Plans of Eco, Low-Carbon and Conventional New Towns in China.', *Habitat International*, 63/2, https://doi.org/10.1016/j.habitatint.2017.03.008, 55–66. (Cited on page 19.)

Garrett, P. and Casimiro, E. (2011), 'Short-term effects of fine particulate matter (PM2.5) and ozone on daily mortality in Lisbon, Portugal', *Environment science Pollution*, 18, https://doi.org/10.1007/s11356-011-0519-z, 1585–92. (Cited on page 84.)

Ghaderi, A., Sanandaji, B. M., and Ghaderi, F. (2017), 'Deep Forecast: Deep Learning-based Spatio-Temporal Forecasting', *arXiv.org*, https://arxiv.org/abs/1707.08110. (Cited on page 51.)

Ghilani, C. D. (2010), 'Adjustment Computations Spatial Data Analysis.', *John Wiley and Sons*, https://doi.org/10.1002/9780470586266. (Cited on page 65.)

Goodchild, M. (2013), 'The quality of big (geo)data', *Dialogues in Human Geography*, 3/17, https://doi.org/10.1177/2043820613513392, 2241–2248. (Cited on page 16.)

Goodchild, M. F. (2007), 'Citizens as sensors: the world of volunteered geography', *GeoJournal*, 69/17, https://doi.org/10.1007/s10708-007-9111-y, 2241–2248. (Cited on page 16.)

Goodfellow, I., Bengio, Y., and Courville, A. (2016), 'Deep Learning', *Cambridge (MA), MIT Press.*, https://www.deeplearningbook.org/contents/intro.html. (Cited on pages 21 and 43.)

Goy, S., Coors, V., and Finn, D. (2021), 'Grouping techniques for building stock analysis: A comparative case study', *Energy and Buildings*, 236, https://doi.org/10.1109/CVPR.2015.7298965, 26. (Cited on page 30.)

Gröger, G. and Plümer, L. (2012), 'CityGML – Interoperable semantic 3D city models.', *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, https://doi.org/10.1016/j.isprsjprs.2012.04.004, 12–33. (Cited on page 18.)

Guilherme, A. B. (2007), 'Time series prediction with the self-organising map', *Springer, In perspective of Neural-symbol integration*, https://doi.org/10.1007/978-3-540-73954-8_6, 135–158. (Cited on pages 36 and 105.)

Guyon, E., Gunn, S., Nikravesh, M., and L., Z. (2006), 'Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing.', *Springer*, https://www.springer.com/gp/book/9783540354871. (Cited on page 29.)

Hao, C. M., Janetzko, H., Mittelstaedt, S., Hill, W., Dayal, U., Keim, D., Marwah, M., and Sharma, R. (2011), 'A visual analytic approach for peak preserving predictions of large seasonal time series', *In computer graphics forum*, 123, https://doi.org/10.1111/j.1467-8659.2011.01918.x, 52–55. (Cited on pages 3, 36, 37, 105, and 106.)

Harbola, S. and Coors, V. (2018), 'Geo-Visualisation and Visual Analytics for Smart Cities: A Survey', *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-4/W11, https://doi.org/10.5194/isprs-archives-XLII-4-W11-11-2018, 11–18. (Cited on pages 5, 15, and 16.)

Harbola, S. and Coors, V. (2019a), 'One Dimensional Convolutional Neural Network Architectures for Wind Prediction', *Energy Conversion and Management*, 195, https://doi.org/10.1016/j.enconman.2019.05.007, 70–75. (Cited on page 5.)

Harbola, S. and Coors, V. (2019b), 'Comparative analysis of LSTM, RF and SVM Architectures for Predicting Wind Nature for smart city planning',

ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-4/W9, https://doi.org/10.5194/isprs-annals-IV-4-W9-65-2019, 65–70. (Cited on page 5.)

Harbola, S. and Coors, V. (2019c), 'Convolutional Neural Network Architectures for Wind Analysis,', *EAWE PhD Seminar 2019 29-31 Oct 2019 Nantes, France*, https://eawephd2019.sciencesconf.org/285035,. (Cited on page 5.)

Harbola, S. and Coors, V. (2020), 'Seasonality Deduction Platform : For PM_{10}, $PM_{2.5}$, NO, NO_2 and O_3 in Relationship with Wind Speed and Humidity', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, VI-4/W2, https://doi.org/10.5194/isprs-annals-VI-4-W2-2020-71-2020, 71–78. (Cited on page 7.)

Harbola, S. and Coors, V. (2021a), 'Deep learning model for wind forecasting', *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 10, https://doi.org/10.1007/s41064-021-00185-6. (Cited on page 5.)

Harbola, S. and Coors, V. (2021b), 'An Interactive Platform For Environmental Sensors Data Analyses', *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci*, VIII-4/W1-2021, https://doi.org/10.5194/isprs-annals-VIII-4-W1-2021-57-2021, 57–64. (Cited on page 8.)

Harbola, S., Koch, S., Ertl, T., and Coors, V. (2021a), 'Air Quality Temporal Analyser: Interactive temporal analyses with visual predictive assessments', *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, https://doi.org/10.2312/envirvis.20211083. (Cited on page 8.)

Harbola, S., Storz, M., and Coors, V. (2021b), *Augment Reality for Windycities:3D Visualisation of future wind nature analysis in city planning* (Springer, (To appear, accepted on 2020- July -20)). (Cited on page 7.)

Hart, K. J. (2006), 'Environmental sensor networks: a revolution in the earth system science?', *Earth-Science Reviews*, 78. (Cited on pages 37 and 106.)

Hashem, I. A. T., Chang, V., Anuar, N. B., Adewole, K., Yaqoob, I., Gani, A., Ahmed, E., and Chiroma, H. (2016), 'The role of big data in smart city', *International Journal of Information Management*, 36/5, https://www.sciencedirect.com/science/article/pii/S0268401216302778, 748–758. (Cited on page 20.)

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008), 'ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning', *International Joint Conference on Neural Networks*, https://doi.org/10.1109/IJCNN.2008.4633969. (Cited on page 71.)

He, K., Zhang, X., Ren, S., and Sun, J. (2016), 'Deep residual learning for image recognition', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, https://doi.org/10.1109/CVPR.2016.90, 770–778. (Cited on page 78.)

Hetland, M. and Saetrom, P. (2005), 'Evolutionary rule mining in time series databases', *Machine learning*, https://doi.org/10.1007/s10994-005-5823-8, 135–158. (Cited on pages 36 and 105.)

Hien, D. P., Bac, T. V., Tham, C. H., Nhan, D. D., and Vinh, D. L. (2002), 'Influence of meteorological conditions on Pm2.5 and Pm2.5-10 concentrations during the monsoon season in Hanoi, Vietnam', *Atmospheric Environment*, 36, https://doi.org/10.1016/S1352-2310(02)00295-9, 3473–3484. (Cited on page 84.)

Hochheiser, H. and Shneiderman, B. (2004), 'Dynamic query tools for time series datasets:timebox widgets for interactive exploration', *Inforamtion Visualisation*, https://doi.org/10.1057/palgrave.ivs.9500061, 1–18. (Cited on pages 36 and 105.)

Hochreiter, S. and Schmidhuber, J. (1997), 'Long Short Term Memory', *Neural Computation*, 9/8, https://doi.org/10.1162/neco.1997.9.8.1735, 1–32. (Cited on page 68.)

Hofmann, H., Follett, L., Majumder, M., and Cook, D. (2012), 'Graphical Tests for Power Comparison of Competing Designs', *IEEE Transactions on Visualization and Computer Graphics*, 18/12, https://doi.org/10.1109/TVCG.2012.230„ 2441–2448. (Cited on page 14.)

Holliman, N., Turner, M., Dowsland, S., Cloete, R., and Picton, T. (2017), 'Designing a cloud-based 3D visualization engine for smart cities', *Society for Imaging Science and Technology*, 69/17, https://www.ingentaconnect.com/contentone/ist/ei/2017/00002017/00000005/art00024?crawler=true&mimetype=application/pdf, 172–178. (Cited on page 16.)

Horvitz, E. (1999), 'Principles of mixed initiative user interfaces', *In Proceedings of the ACM Conference on Human Factors in Computing Systems*, 53, https://doi.org/10.1145/302979.303030, 59–67. (Cited on pages 37, 105, 111, and 127.)

Horvitz, E. (2007), 'Reflections on challenges and promises of mixed intiative interaction', *AI Magazine*, 53, https://doi.org/10.1609/aimag.v28i2.2036, 59–67. (Cited on pages 37, 106, and 121.)

Huang, C. and Liang, S. (2014), 'A sensor data mediator bridging the OGC Sensor Observation Service (SOS) and the OASIS Open Data Protocol (OData)', *Society for Imaging Science and Technology*, 185/12, https://doi.org/10.1080/19475683.2014.942795, 279–293. (Cited on page 16.)

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, Q. K. (2018), 'Densely connected convolutional networks', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, https://doi.org/https://arxiv.org/abs/1608.06993, 2261–2269. (Cited on page 78.)

Isenberg, P., Heimerl, F., Koch, S., Isenberg, T., Xu, P., Stolper, C. D., Sedlmair, M., Chen, J., Moller, T., and Stasko, J. (2017), 'vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications', *IEEE Transactions on Visualization and Computer Graphics*, 23, https://doi.org/10.1109/TVCG.2016.2615308, 2199–2206. (Cited on pages 2 and 35.)

Jasen, N. A. H., Fischer, P., Marra, A. C., and Cassee, F. R. (2013), 'Short-term effects of PM2.5, PM10 and PM2.5-10 on daily mortality in The Netherlands', *Total Environment*, 463, https://doi.org/10.1016/j.scitotenv.2013.05.062, 20–36. (Cited on pages 84 and 120.)

Javed, W., Mcdonnel, B., and Elmqvist, N. (2010), 'Graphical perception of multiple time series', *IEEE Transactions on Visualization and Computer Graphics*, https://doi.org/10.1109/TVCG.2010.162. (Cited on page 112.)

Jeong, Y., Jeong, M., and Omitaaomu, O. (2011), 'Weighted dynamic time warping for time series classification', *Pattern recognition*, https://doi.org/10.1016/j.patcog.2010.09.022. (Cited on pages 36 and 105.)

Johnson, C. (2004), 'Top scientific visualization research problems', *IEEE Computer Graphics and Applications*, 24/4, https://ieeexplore.ieee.org/document/1310205, 13–17. (Cited on page 14.)

Jung, D., Jung, W., Kim, B., Lee, S., Rhee, W., and Ahn, J. H. (2019), 'Restructuring Batch Normalization to Accelerate CNN Training', *ArXiv*, abs/1807.01702, https://arxiv.org/abs/1807.01702. (Cited on pages 35, 57, and 71.)

Jursa, R. and Rohrig, K. (2008), 'Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models', *International Journal of Forecasting*, 24, https://doi.org/10.1016/j.ijforecast.2008.08.007, 694–709. (Cited on page 34.)

Kalamaras, I., Zamichos, A., Salamanis, A., Drosou, A., Kehagias, D. D., Margaritis, G., Papadopoulos, S., and Tzovaras, D. (2018), 'An Interactive Visual Analytics Platform for Smart Intelligent Transportation Systems Management', *IEEE Transactions on Intelligent Transportation Systems*, 19/2, https://doi.org/10.1109/TITS.2017.2727143, 487–496. (Cited on page 20.)

Kang, A., Tan, Q., Yuan, X., Lei, X., and Yuan, Y. (2017), 'Short-Term Wind Speed Prediction Using EEMD-LSSVM Model', *Advances in Meteorology*, https://doi.org/10.1155/2017/6856139. (Cited on page 34.)

Kapoor, A., Lee, B., Tan, D., and Horvitz, E. (2012), 'Performance and preferences: interactive refinement of Machine learning procedures', *Proceedings of the AAAI Conference on AI*, 2, https://dblp.org/rec/conf/aaai/KapoorLTH12a.html, 59–67. (Cited on pages 37 and 106.)

Karpathy, A., Johnson, J., and Fei-Fei, L. (2016), 'Visualizing and Understanding Recurrent Networks', *International Conference on Learning Representations 2016*, https://arxiv.org/abs/1506.02078, 1–11. (Cited on page 68.)

Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, F. (2010), 'Mastering The Information Age: Solving Problems with Visual Analytics', *Goslar : Eurographics Association*, https://diglib.eg.org/handle/10.2312/14803. (Cited on pages 17 and 18.)

Kitchin, R. (2011), 'Big data and human geography: Opportunities, challenges and risks', *Dialogues in Human Geography*, 17/12, https://doi.org/10.1177/2043820613513388, 2241–2248. (Cited on page 14.)

KNMI (2020), 'National knowledge institute for weather, climate and seismology KNMI', *The Royal Netherlands Meteorological Institute*, http://projects.knmi.nl/klimatologie/frequentietabellen/uur_freq.cgi.htm. (Cited on page 41.)

Kohlhammer, J., Ruppert, T., Davey, J., Mansmann, F., and Keim, D. (2010), 'Information Visualisation and Visual Analytics for Governance and Policy Modelling', *Eur. Comm. Futur. Res. ICT Gov. Policy Model. Crossroad -*

A Particip. Roadmap ICT Res. Electron. Gov. Policy Model, 8/2, https://bib.dbvis.de/uploadedFiles/306.pdf, 20–28. (Cited on page 20.)

Komninos, N., Pallot, M., and Schaffers, H. (2013), 'Special Issue on Smart Cities and the Future Internet in Europe', *Journal of the Knowledge Economy*, 4/14, https://doi.org/10.1007/s13132-012-0083-x, 119–134. (Cited on page 13.)

Krause, J., Perer, A., and Ng, K. (2016), 'Integrating with predictions: Visual inspection of black-box machine learning models', *Association for Computing Machinery ACM*, https://doi.org/10.1145/2858036.2858529. (Cited on pages 36 and 104.)

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012), 'ImageNet Classification with Deep Convolutional Neural Networks', *Stanford Vision Lab*, https://doi.org/10.1145/3065386. (Cited on pages 26, 35, and 78.)

Kumar, S. and Prakash, A. (2016), 'Planning for Sustainable Cities? A Comparative Content Analysis of the Master Plans of Eco, Low-Carbon and Conventional New Towns in China.', *Role of Big Data and Analytics in Smart Cities*, 63/2, https://doi.org/10.21275/v5i2.nov161007, 55–66. (Cited on page 19.)

Kuo, C. C. (2016), 'Understanding convolutional neural networks with a mathematical model', *Journal of Visual Communication and Image Representation*, 41, https://doi.org/10.1016/j.jvcir.2016.11.003, 406–413. (Cited on page 35.)

Kurkcu, A., Miranda, F., Ozbay, K., and Silva, C. (2017), 'Data visualization tool for monitoring transit operation and performance.', *IEEE Int. Conf. Model. Technol. Intell. Transp. Syst.*, https://doi.org/10.1109/MTITS.2017.8005584, 598–603. (Cited on page 121.)

Kusiak, A. and Zhang, Z. (2010), 'Short-horizon prediction of wind power: A data-driven approach', *Energy Conversion, IEEE Transactions on*, 25, https://doi.org/10.1109/TEC.2010.2043436, 1112–1122. (Cited on page 34.)

Kusiak, A., Zheng, H., and Song, Z. (2009a), 'Models for monitoring wind farm power', *Renewable Energy*, 34, https://doi.org/10.1016/J.RENENE.2008.05.032, 583–590. (Cited on page 34.)

Kusiak, A., Zheng, H., and Song, Z. (2009b), 'Short-Term Prediction of Wind Farm Power: A Data Mining Approach', *Energy Conversion, IEEE Transactions on*, 24, https://doi.org/10.1109/TEC.2008.2006552, 125–136. (Cited on page 34.)

Köthur, P., Sips, M., Kuhlmann, J., and Dransch, D. (2012), 'Visualization of Geospatial Time Series from Environmental Modeling Output', *Eurographics Conference on Visualization (EuroVis)*, http://dx.doi.org/10.2312/PE/EuroVisShort/EuroVisShort2012/115-119. (Cited on page 3.)

Lamba, A., Cassey, P., Segaran, R., and Koh, P. (2019), 'Deep learning for environmental conservation', *Current Biology*, 29/19, https://doi.org/10.1016/j.cub.2019.08.016, R977–R982. (Cited on page 2.)

Lara, A., Costa, E., Furlani, X., and Yigitcanlar, T. (2016), 'Smartness that matters: towards a comprehensive and human-centred characterisation of smart cities', *Journal of Open Innovation: Technology, Market, and Complexity*, 8/2, https://doi.org/10.1186/s40852-016-0034-z, 10 pp.–. (Cited on page 20.)

Lawan, M. S., Abidin, A. W., Chai, Y. W., Baharun, A., and Masri, T. (2014), 'Different Models of Wind Speed Prediction; A Comprehensive Review', *International Journal of Scientific & Engineering Research*, 5/8, https://doi.org/10.1.1.428.9347, 1760–1768. (Cited on page 33.)

Liao, T. W. (2005), 'Clustering of time series data-a survey', *Pattern recognition*, 38/11, https://doi.org/10.1016/j.patcog.2005.01.025. (Cited on pages 36 and 105.)

Liu, H. and Motoda, H. (1998), 'Feature Extraction, Construction and Selection: A Data Mining Perspective.', *Springer*, https://www.springer.com/gp/book/9780792381969. (Cited on pages 29 and 30.)

Liu, H., Mi, X., and Li, Y. (2018), 'Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network', *Energy Conversion and Management*, 166, https://doi.org/10.1016/j.enconman.2018.04.021, 120–131. (Cited on pages 2, 34, 35, 61, 62, and 79.)

Liu, S. W., Liu, M., and Zhu, J. (2017), 'Towards better analysis of machine learning models: A visual analytics perspective', *Vis. Informatics.*, https://arxiv.org/abs/1702.01226, 598–603. (Cited on page 121.)

Lloyd, D. and Dykes, J. (2011), 'Human-Centered Approaches in Geovisualization Design: Investigating Multiple Methods Through a Long-Term Case Study', *IEEE Transactions on Visualization and Computer Graphics*, 17/12, https://doi.org/10.1109/TVCG.2011.209„ 2498–2507. (Cited on page 14.)

Long, J., Shelhamer, E., and Darrell, T. (2015), 'Fully Convolutional Networks for Semantic Segmentation', *International Journal of Digital Earth*, https://doi.org/10.1109/CVPR.2015.7298965. (Cited on page 35.)

Lorenc, A. C. (1986), 'Analysis methods for numerical weather prediction', *Royal meteorological society quarterly journal*, https://doi.org/10.1002/qj.49711247414. (Cited on pages 36 and 105.)

Louka, P., Galanis, G., Siebert, N., Kariniotakis, G., Katsafados, P., Pytharoulis, I., and Kallos, G. (2008), 'Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering', *Journal of Wind Engineering and Industrial Aerodynamics*, 96, https://doi.org/10.1016/j.jweia.2008.03.013, 2348–2362. (Cited on page 34.)

Lu, Y., Krueger, R., Thom, D., Koch, S., Ertl, T., and Maciejewski, R. (2014), 'Integrating Predictive analytics and social media', *IEEE symposium on visual analytics science and technology*, 123, https://doi.org/10.1109/VAST.2014.7042495, 52–55. (Cited on pages 36 and 105.)

Luftdata-se-Stuttgart (2020), 'Stuttgart datasets luftdata.se', *luftdaten.info – Feinstaub selber messen*, https://www.luftdaten.info. (Cited on pages 107 and 129.)

MacEachren, A. M. and Kraak, M. (2001), 'Research challenges in geovisualization', *Cartography and Geographic Information Science*, 28, https://www.tandfonline.com/doi/abs/10.1559/152304001782173970, 3–12. (Cited on page 14.)

MacEachren, A. M. and Kraak, M. (2011), 'Exploratory cartographic visualisation: Advancing the agenda. Map Read', *The Map Reader: Theories of Mapping Practice and Cartographic Representation*, 23, https://doi.org/10.1002/9780470979587.ch11, 83–88. (Cited on page 14.)

MacEachren, A. M., Roth, R. E., O'Brien, J., Swingley, D., and Gahegan, M. (2012), 'Visual Semiotics & Uncertainty Visualization: An Empirical Study', *IEEE Transactions on Visualization and Computer Graphics*, 18, https://doi.org/10.1109/TVCG.2012.279, 1–10. (Cited on page 14.)

MacEachren, A. M., Robinson, A., and Hopper, S. (2013), 'Visualizing geospatial information uncertainty: What we know and what we need to know', *Cartography and Geographic Information Science*, 32, https://doi.org/10.1559/1523040054738936, 139–160. (Cited on page 13.)

Maciejewski, R., Rudolph, S., Hafen, R., Abusalah, A., Yakout, M., Ouzzani, M., Cleveland, W. S., Grannis, S. J., and Ebert, D. S. (2010), 'A Visual Analytics Approach to Understanding Spatiotemporal Hotspots', *IEEE Transactions on Visualization and Computer Graphics*, 16/2, https://ieeexplore.ieee.org/document/5226628, 205–220. (Cited on page 14.)

Malik, A., MAciejewski, R., Jang, Y., Huang, W., Elmquivst, N., and Ebert, D. (2012), 'A correlative analysis process in a visual analytic environment', *IEEE symposium on information visualisation*, 123, https://doi.org/10.1109/VAST.2012.6400491, 52–55. (Cited on pages 37 and 106.)

Marović, I., Sušanj, I., and Ožanić, N. (2017), 'Development of ANN model for wind speed prediction as a support for early warning system', *Complexity*, https://doi.org/10.1155/2017/3418145. (Cited on page 33.)

Marsal-Llacuna, M. (2015), 'Lessons in urban monitoring taken from sustainable and livable cities to better address the Smart Cities initiative.', *Technological Forecasting and Social Change an International Journal*, 90, https://doi.org/10.1016/j.techfore.2014.01.012, 611. (Cited on page 18.)

Martínez-Arellano, G., Nolle, L., Cant, R., Lotfi, A., and Windmill, C. (2014), 'Characterisation of Large Changes in Wind Power for the Day-Ahead Market Using a Fuzzy Logic Approach', *KI - Künstliche Intelligenz*, 28, https://doi.org/10.1007/s13218-014-0322-3, 239–253. (Cited on page 34.)

McInnes, L. and Healy, J. (2017), 'Accelerated Hierarchical Density Clustering', *International Conference on Data Mining Workshops (ICDMW). Proceedings of IEEE 2017*, https://arxiv.org/abs/1705.07321. (Cited on page 125.)

Memisevic, R., Zach, C., Hinton, G., and Pollefeys, M. (2010), 'Gated Softmax Classification', *Advances in Neural Information Processing Systems*, https://proceedings.neurips.cc/paper/2010/hash/5737c6ec2e0716f3d8a7a5c4e0de0d9a-Abstract.html, 1–9. (Cited on page 56.)

Miller, T. (2017), 'Explanation in Artificial Intelligence: Insights from the Social Sciences', *Artificial Intelligence*, 267, https://doi.org/10.1016/j.artint.2018.07.007. (Cited on page 21.)

Miranda, M. and Dunn, R. (2006), 'One-hour-ahead wind speed prediction using a Bayesian methodology', *IEEE Power Engineering Society General*

Meeting, https://doi.org/10.1109/PES.2006.1709479, 1–6. (Cited on page 34.)

Monfared, M., Rastegar, H., and Kojabadi, H. M. (2009), 'A new strategy for wind speed forecasting using artificial intelligent methods', *Renewable Energy*, 34, https://doi.org/10.1016/j.renene.2008.04.017, 845–848. (Cited on page 34.)

Nair, V. and Hinton, G. (2010), 'Rectified Linear Units Improve Restricted Boltzmann Machines', *Proceedings of the International Conference on Machine Learning (ICML-10)*, 27/6, https://dl.acm.org/doi/10.5555/3104322.3104425, 1–7. (Cited on page 71.)

Navarra, C., Opach, T., Vrotsou, K., Joling, A., Wilk1, J., and Neset, T. (2020), 'Visual Exploration of Climate-Related Volunteered Geographic Information', *Workshop on Visualisation in Environmental Sciences (EnvirVis) (2020)*, 123, https://doi.org/10.2312/envirvis.20201092, 52–55. (Cited on pages 36 and 105.)

Nga, D., See, O., Quang, D., Xuen, C., and Chee, L. (2012), 'Visualization Techniques in Smart Grid.', *Visualization Techniques in Smart Grid*, 3/3, https://doi.org/10.4236/sgre.2012.33025, 175–185. (Cited on page 20.)

Nielsen, M. A. (2015), 'Neural Networks and Deep Learning', *Springer*, http://neuralnetworksanddeeplearning.com/. (Cited on pages 24, 27, and 28.)

Noon, C. and Hankins, C. (2001), 'Spatial data visualization in healthcare: supporting a facility location decision via GIS-based market analysis', *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, https://doi.org/10.1109/HICSS.2001.926573, 10 pp.–. (Cited on page 20.)

Otto, M., Germer, T., Hege, H.-C., and Theisel, H. (2010), 'Uncertain 2D Vector Field Topology', *Computer Graphics Forum*, 29/2, https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01604.x, 347–356. (Cited on page 13.)

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. (2018), 'Image Transformer', *Proceedings of the 35 th International Conference on Machine Learning, Stockholm, Sweden*, https://arxiv.org/abs/1802.05751. (Cited on page 125.)

Paul and Murphy, A. (2009), ‘Data Mining’, *Distances between Clustering, Hierarchical Clustering*, https://docplayer.net/10404659-Distances-between-clustering-hierarchical-clustering.html, 36–350. (Cited on pages 86 and 125.)

Pedamonti, D. (2018), ‘Comparison of non-linear activation functions for deep neural networks on MNIST classification task’, *arXiv.org*, https://arxiv.org/abs/1804.02763. (Cited on pages 57 and 71.)

Pirolli, P. and Card, S. K. (2005), ‘The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis’, *Proceedings of International Conference on Intelligence Analysis*, https://www.researchgate.net/publication/215439203_The_sensemaking_process_and_leverage_points_for_analyst_technology_as_identified_through_cognitive_task_analysis, 2–4. (Cited on page 17.)

Prandi, F., Amicis, R., Piffer, S., Soavea, M., Cadzowb, S., B., G., and D’Hont, D. (2013), ‘USING CITYGML TO DEPLOY SMART-CITY SERVICES FOR URBAN ECOSYSTEMS.’, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 11/8, https://doi.org/10.5194/isprsarchives-XL-4-W1-87-2013, 87–92. (Cited on page 16.)

Prieto, D. F. (2013), ‘UCIV 4 PLANNING: A USER-CENTERED APPROACH FOR THE DESIGN OF INTERACTIVE VISUALIZATIONS TO SUPPORT URBAN AND REGIONAL PLANNING’, *IADIS International Journal on Computer Science and Information Systems*, 8/2, www.iadisportal.org/ijcsis/papers/2013160203.pdf, 27–39. (Cited on page 19.)

Qi, R. C., Su, H., Niessner, M., Dai, A., Yan, M., and Guibas, J. L. (2016), ‘Volumetric and Multi-View CNNs for Object Classification on 3D Data’, *Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings of the IEEE*, https://doi.org/10.1109/CVPR.2016.609. (Cited on pages 35 and 61.)

Ramathan, A., Dykes, J., and Wood, J. (2013), ‘Framework for studying spatially ordered treemaps’, *International Cartographic Conference: From Pole to Pole*, 26/4, https://openaccess.city.ac.uk/id/eprint/2609, 25 – 30. (Cited on page 13.)

Reed, C. T., Fiffick, A. S., and Sawyers, R. D. (2011), ‘Wind Data Analysis and Performance Predictions for a 400-kW Turbine in Northwestern Ohio’, *American Society for Engineering Education, ASEE - 2011. Proceedings*,

http://people.cst.cmich.edu/yelam1k/asee/proceedings/2011/data/17-162-3-dr.pdf, 1–9. (Cited on page 33.)

Sacha, D., Stoffel, A., Kwon, B. C., Ellis, G., and Keim, D. A. (2014), 'Knowledge Generation Model for Visual Analytics', *Visualization and Computer Graphics, IEEE Transactions on*, 20/12, 1604– 1613. (Cited on page 17.)

Sacha, D., Senaratne, H., Kwon, C., Ellis, G., and Keim, A. (2016), 'The role of uncertainty, awareness and trust in visual analytics', *IEEE Transactions on Visualization and Computer Graphics*, 22/1, https://doi.org/10.1109/TVCG.2015.2467591. (Cited on pages 36 and 105.)

Sanchez, A. and Rivera, W. (2017), 'Big Data Analysis and Visualization for the Smart Grid', *IEEE International Congress on Big Data (BigData Congress)*, 3/3, https://doi.org/10.1109/BigDataCongress.2017.59, 414–418. (Cited on page 20.)

Sapronova, A., Johannsen, K., Thorsnes, E., Meissner, C., and Mana, M. (2016), 'Deep learning for wind power production forecast', *CEUR Workshop - 2016, Proceedings*, 1818, http://ceur-ws.org/Vol-1818/paper3.pdf, 28–33. (Cited on page 34.)

Shneiderman, B. (1996), 'The eye have it: a task by data type taxonomy for information visualization.', *Proceedings of IEEE Visual Languages, College Park, Maryland, pp. 336–343 (1996)*, 9, https://www.mat.ucsb.edu/~g.legrady/academic/courses/11w259/schneiderman.pdf, 3–12. (Cited on pages 12, 35, 104, 111, and 127.)

Shuyang, D., Pandey, M., and Xing, C. (2017), 'Modeling approaches for time series forecasting and anomaly detection', *Semantic Scholar*, http://cs229.stanford.edu/proj2017/final-reports/5244275.pdf, 8–13. (Cited on page 51.)

Singh, D., Vishnu, C., and C, K. M. (2016), 'Visual Big Data Analytics for Traffic Monitoring in Smart City', *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 3/3, https://doi.org/10.1109/ICMLA.2016.0159, 18–20. (Cited on page 20.)

Southworth, F. and Peterson, B. E. (2000), 'Intermodal and international freight network modeling', *Transportation Research Part C: Emerging Technologies*, 8/1, https://doi.org/10.1016/S0968-090X(00)00004-8, 147–166. (Cited on page 14.)

Srivastava, N., Hinton, G., Sutskever, I., and Salakhutdinov, R. (2014), 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting',

Journal of Machine Learning Research, https://dl.acm.org/doi/10.5555/2627435.2670313. (Cited on pages 68 and 71.)

Stadtentwicklung.berlin.de (2021), 'Traffic-Related Air Pollution Along Streets 2015', *Environmental Atlas Berlin*, https://www.berlin.de/umweltatlas/en/air/traffic-related-emissions-and-immissions/2015/maps/artikel.981256.en.php. (Cited on page 117.)

Stadtklima-News (2021), 'Stadtklima Stuttgart News', *Stadklima. Sututtgart*, https://www.stadtklima-stuttgart.de/index.php?info_news. (Cited on pages 118 and 119.)

Stadtklima-Stuttgart (2021), 'Stuttgart datasets website Stadtklima-Stuttgart', *Stadtklima-Stuttgart*, https://www.stadtklima-stuttgart.de/index.php?start.htm. (Cited on pages 41, 87, 108, and 129.)

Stefan, M., Lopez, J. G., Andreasen, M. H., and Olsen, R. L. (2017), 'Visualization Techniques for Electrical Grid Smart Metering Data: A Survey', *IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*, https://doi.org/10.1109/BigDataService.2017.26, 165–171. (Cited on pages 16 and 20.)

Stratigea, A., Kyriakides, E., and Nicolaides, C. (2017), 'Smart cities in the Mediterranean', *Springer*, 23/31-58, https://doi.org/10.1007/978-3-319-54558-5. (Cited on page 121.)

Su, H., Maji, S., Kalogerakis, E., and miller, E. (2015), 'Multi-view Convolutional Neural Networks for 3D Shape Recognition', *International Conference on Computer Vision, Proceedings of the IEEE*, https://doi.org/10.1109/ICCV.2015.114, 945–953. (Cited on pages 56 and 110.)

Sugumaran, V. and Sugumaran, R. (2013), 'Web-based Spatial Decision Support Systems (WebSDSS): Evolution, Architecture, Examples and Challenges.', *Communications of the Association for Information Systems*, 19/8, https://doi.org/10.17705/1CAIS.01940, 87–92. (Cited on page 17.)

Sun, G., Wu, Y., Liang, R., and Liu, S. (2013), 'A Survey of Visual Analytics Techniques and Applications: State-of-the-Art Research and Future Challenges', *Journal of Computer Science and Technology*, 28/12, https://doi.org/10.1007/s11390-013-1383-8, 852–867. (Cited on pages 15 and 121.)

Sun, Y. and Li, S. (2016), 'Real-time collaborative GIS: A technological review', *ISPRS Journal of Photogrammetry and Remote Sensing*, 115, https://www.sciencedirect.com/science/article/pii/S092427161500221X, 143–152. (Cited on pages 15 and 121.)

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015), 'Going deeper with convolutions', *Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings of the IEEE*, 07-12-June, https://doi.org/10.1109/CVPR.2015.7298594, 1–9. (Cited on page 35.)

Tan, S., Yang, J., Yan, Y., Lee, C., Hashim, H., and Chen, B. (2017), 'A holistic low carbon city indicator framework for sustainable development', *Applied Energy*, 185, https://doi.org/10.1016/j.apenergy.2016.03.041, 1919–1930. (Cited on page 19.)

Tarade, R. S. and Katti, P. K. (2011), 'A comparative analysis for wind speed prediction', *International Conference on Energy, Automation and Signal, ICEAS - 2011. Proceedings*, https://doi.org/10.1109/ICEAS.2011.6147167, 556–561. (Cited on page 33.)

Thomas, J. and Kielman, J. (2009), 'Challenges for visual analytics', *Information Visualization*, 8/4, https://www.uni-konstanz.de/mmsp/pubsys/publishedFiles/KeMaSc06.pdf, 309–314. (Cited on page 17.)

Thomas, J. J. and Cook, A. K. (2005), 'Illuminating the Path: The Research and Development Agenda for Visual Analytics', *IEEE Computer Society Press*, 54, https://www.hsdl.org/?abstract&did=485291. (Cited on pages 2, 17, 18, and 35.)

Thomson, J., Hetzler, E., MacEachren, A., Gahegan, M., and Pavel, M. (2005), 'A typology for visualizing uncertainty', *Proceedings Volume 5669, Visualization and Data Analysis*, 5669/4, https://doi.org/10.1117/12.587254, 58–62. (Cited on page 14.)

Treiber, N., Heinermann, J., and Kramer, O. (2016), 'Wind Power Prediction with Machine Learning', *Studies in Computational Intelligence*, 645, https://doi.org/10.1007/978-3-319-31858-5_2. (Cited on page 34.)

Trindade, E. P., Phoebe, M., Hinnig, F., Costa, E. M., Marques, M. J., Bastos, R., and Yigitcanlar, B. (2017), 'Sustainable development of smart cities: a systematic review of the literature', *Journal of Open Innovation: Technology, Market, and Complexity*, 3/11, https://doi.org/10.1186/s40852-017-0063-2„ 2310–2316. (Cited on page 14.)

Tsolakis, N. and Anthopoulos, L. (2015), 'Eco-cities: An integrated system dynamics framework and a concise research taxonomy', *Sustainable Cities and Society*, 17, https://www.sciencedirect.com/science/article/pii/S2210670715000220, 1–14. (Cited on page 20.)

Tunio, S., Kazi, H., and Qureshi, S. (2017), 'Customization of Graphical Visualization for Health Parameters in Health Care Applications', *International Journal of Advanced Computer Science and Applications(IJACSA)*, 8/9, https://doi.org/10.14569/IJACSA.2017.080913, 14–18. (Cited on page 20.)

Vargas, L., Paredes, G., and Bustos, G. (2010), 'Data mining techniques for very short term prediction of wind power', *IREP Symposium Bulk Power System Dynamics and Control - VIII (IREP)*, 8/6, https://doi.org/10.1109/IREP.2010.5563273, 1–7. (Cited on pages 33 and 34.)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, L. J.and Jones, Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017), 'Attention Is All You Need', *Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.*, https://arxiv.org/abs/1706.03762. (Cited on page 125.)

Venugopal, K. R., Srinivasa, G. K., and Patnaik, L. M. (2011), 'Fuzzy based neuro genetic algorithm for stock market prediction', *Soft computing for data mining applications*, 123, https://link.springer.com/chapter/10.1007/978-3-642-00193-2_7, 52–55. (Cited on pages 36 and 105.)

Vilone, G. and Longo, L. (2020), 'Explainable Artificial Intelligence: a Systematic Review', *ArXiv*, https://arxiv.org/abs/2006.00093, 84–92. (Cited on page 21.)

Vladislavleva, E., Friedrich, T., Neumann, F., and Wagner, M. (2013), 'Predicting the energy output of wind farms based on weather data: Important variables and their correlation', *Renewable Energy*, 50/6, https://doi.org/10.1016/j.renene.2012.06.036, 236–243. (Cited on page 34.)

Wallace, J. M. and Hobbs, P. V. (1977), 'Academic press: New York', *Atmospheric science, an introductory survey*, 467, https://ci.nii.ac.jp/naid/10003424005/. (Cited on pages 84 and 120.)

Wang, H., Li, G., Wang, G., Peng, J., Jiang, H., and Liu, Y. (2017a), 'Deep learning based ensemble approach for probabilistic wind power forecasting', *Applied Energy*, 188, https://doi.org/10.1016/j.apenergy.2016.11.111, 56–70. (Cited on pages 34 and 35.)

Wang, W., De, S., Zhou, Y., Huang, X., and Moessner, K. (2017b), 'Distributed sensor data computing in smart city applications', *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, https://doi.org/10.1109/WoWMoM.2017.7974338, 1–5. (Cited on page 20.)

Ward, J. H. (1963), 'Hierarchical Grouping to Optimize an Objective Function', *Journal of the American Statistical Association*, 58, https://doi.org/10.1080/01621459.1963.10500845, 236–244. (Cited on pages 30 and 86.)

Weber, M., Alexa, M., and Muller, W. (2011), 'Visualising time series on spirals', *IEEE symposium on information visualisation*, 123, https://doi.org/10.1109/INFVIS.2001.963273, 52–55. (Cited on pages 36 and 105.)

Wu, N., Green, B., Ben, X., and Banion, S. (2020), 'Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case', *International Conference on Machine Learning, Vienna, Austria*, https://arxiv.org/abs/2001.08317. (Cited on page 125.)

Xie, N., Ras, G., Gerven, M., and Doran, D. (2020), 'Explainable Deep Learning: A Field Guide for the Uninitiated', *ArXiv*, https://arxiv.org/abs/2004.14545, 84–92. (Cited on pages 4 and 21.)

Xie, S., Girshick, R., and Doll, P. (2017), 'Aggregated Residual Transformations for Deep Neural Networks', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, https://doi.org/10.1109/CVPR.2017.634, 2261–2269. (Cited on page 78.)

Yesilbodak, M., Sagiroglu, S., and Colak, I. (2017), 'A novel implementation of kNN classifier based on multi-tupled meteorological input data for wind power prediction', *Energy Conversion and Management*, 135, https://doi.org/10.1016/j.enconman.2016.12.094, 434–444. (Cited on page 34.)

Yesilbudak, M., Sagiroglu, S., and Colak, I. (2013), 'A new approach to very short term wind speed prediction using k-nearest neighbor classification', *Energy Conversion and Management*, 69, https://doi.org/10.1016/j.enconman.2013.01.033, 77–86. (Cited on page 34.)

Yigitcanlar, T. and Kamruzzaman, M. (2018), 'Does smart city policy lead to sustainability of cities?', *Land Use Policy*, 73, https://doi.org/10.1016/j.landusepol.2018.01.034, 49–58. (Cited on page 20.)

Yuan, X., Chen, C., Yuan, Y., Huang, Y., and Tan, Q. (2015), 'Short-term wind power prediction based on LSSVM-GSA model', *Energy Conversion and Management*, 101, https://doi.org/10.1016/j.enconman.2015.05.065, 393–401. (Cited on pages 34 and 51.)

Zhao, J., Chevalier, F., and Balakrishnan, R. (2011a), 'KronoMiner: using multi-foci navigation for the visual exploration of time-series data', *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems*, 123, https://doi.org/10.1145/1978942.1979195, 52–55. (Cited on pages 36 and 105.)

Zhao, J., Chevalier, F., and Balakrishnan, R. (2011b), 'Exploratory analysis of time-series with chronolenses', *IEEE Transactions on Visualization and Computer Graphics*, https://doi.org/10.1109/TVCG.2011.195. (Cited on pages 36 and 105.)

Zhao, Z., Zheng, P., Xu, S., and Wu, X. (2019), 'Object detection with deep learning:A review', *IEEE Transactions on Neural Network and Learning Systems*, https://arxiv.org/abs/1807.05511. (Cited on page 66.)

Zhou, H., Jiang, J. X., and Huang, M. (2011), 'Short-term wind power prediction based on statistical clustering', *IEEE Power Engineering Society General Meeting*, https://doi.org/10.1109/PES.2011.6039233, 1–7. (Cited on page 34.)

All above links were last followed on 2021-July-20