UCL Department of Electronic and Electrical Engineering

IOT Indoor Air Quality Networks for Smart Homes

Vishal Rajagopal

Declaration Of Authorship

I, Vishal Rajagopal declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;

2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;

3. Where I have consulted the published work of others, this is always clearly attributed;

4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;

5. I have acknowledged all main sources of help;

6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;

Signed: ……  ……

Date: 13/12/2024

Abstract

Utilising predictive modelling and innovative data collection methods can yield a comprehensive understanding, thus guiding the enhancement of indoor air quality (IAQ). The central goal of this doctoral study is to construct a customised intelligent IoT system that integrates diverse air quality sensing techniques and data from smart home automation systems. By implementing neural network-based methodologies, the research showcases the system's adeptness in accurately forecasting forthcoming air quality conditions. These projections can facilitate proactive adjustments to household elements, including ventilation, to enhance air quality.

The data collection framework encompasses a wireless sensor node equipped with various strategically positioned sensors within households, complemented by the capacity to gather data from existing building and home automation systems. Initially employing the Long Short-Term Memory Neural Network (LSTM), the study examines the relationships among air quality factors through univariate and multivariate LSTM analyses.

Preliminary findings underscore the effectiveness of the wireless sensor modules in capturing crucial and dependable data for neural network training. The neural network employs this data to construct a dynamic predictive model for anticipated air quality, assuming a continuous influx of real-time air quality data into the system.

Furthermore, this study explores a novel variant of LSTM that integrates a shared hidden state. The primary objective is to facilitate the examination of interconnected prediction data sourced from various locations to identify potential correlations between indoor air quality levels across different sites. The study seeks to explore how these correlations can enhance predictions related to indoor air quality.

In the future, the research will broaden the scope of IAQ data integration by incorporating data from existing building automation systems into the LSTM model. The objective is to identify correlations between controllable aspects of building automation systems and indoor air quality, thus paving the way for further advancements in this domain.

Impact statement

The quality of the air within our living spaces has long been a point of concern, given the significant amount of time people spend indoors. Recognising the potential threats posed by compromised indoor air quality (IAQ), the presented research has taken a groundbreaking leap forward in comprehending and enhancing IAQ. By harnessing the capabilities of advanced predictive modelling coupled with innovative data collection methods, the study is poised to revolutionise our understanding and control of indoor environments.

The neural network-based methodologies, primarily focusing on the Long Short-Term Memory Neural Network (LSTM), stand as a testament to the prowess of modern computational technologies. The reliability and effectiveness of wireless sensor modules, as corroborated by the study, have vast implications. Individuals no longer have to rely on static or periodic reports about IAQ. Instead, real-time and dependable data streaming allows for the dynamic prediction of forthcoming air quality conditions. This shift from a reactive to a proactive stance is crucial. Imagine being able to adjust ventilation systems or other household elements in anticipation of deteriorating air conditions, thereby maintaining optimal living conditions at all times.

Moreover, the study is not just confined to a single household or building. Exploring a novel variant of LSTM that incorporates a shared hidden state delves into the realm of interconnected prediction data sourced from multiple locations. The ability to identify potential correlations between IAQ levels across different sites provides a broader, interconnected understanding of IAQ dynamics. Such insights could pave the way for community or city-wide interventions, optimising IAQ on a much larger scale than previously imagined.

Yet, the research's ambition does not stop there. With plans to integrate data from existing building automation systems into the LSTM model in the future, the study is set to bridge the gap between controllable aspects of building systems and IAQ. This holistic integration is critical for the development of smart cities and communities where every component, from building designs to ventilation systems, works in harmony to ensure the health and well-being of its residents.

In conclusion, as our world steadily transitions towards the age of smart homes and interconnected buildings, the importance of IAQ cannot be overstated. With its innovative approach and promising methodologies, this research not only sets the stage but actively drives us towards a future where living spaces are not just smart but also inherently healthier. By aligning state-of-the-art technology with our intrinsic need for quality air, this study promises a brighter, healthier future for all.

Table of Contents

Table of figures

Table of tables

1 Introduction

Air pollution is not limited to the outdoors but is also present indoors within our households, offices, and schools. Indoor air quality affects the health and well-being of occupants in the building. The concentration of some pollutants can be multiple times higher in the indoor environment, where many of us spend up to 90% of our time [1],[2]. Contrary to what most people think, indoor air quality is not solely caused by the outdoor pollutants leaking into the indoors, but it is a mix of outdoor sources as well as emissions from building materials and furnishings, central heating and cooling systems, humidification devices, moisture processes, electronic equipment, products for household cleaning, pets, and the behaviour of building occupants [3], [4]. As the sources of indoor pollutants are usually very localised and vary from different households, [5] this study aims to build a system that can analyse the sources of Indoor Air Quality(IAQ)as well as use the collected data to predict future IAQ values where the system is installed and using this information to improve the indoor air quality[3], [4], [6].

This thesis proposes an indoor air quality monitor system which can predict air quality through an improved LSTM algorithm to work seamlessly with the automation system in a smart home. The improved LSTM algorithm incorporates multiple sites into a single model to attempt to overcome the localised nature of applying LSTM to air quality data.

1.1 Indoor Air Quality

There are multiple methods to define IAQ. However, in general, IAQ is characterised by the depictions of concentrations of pollutants that may adversely affect the health and comfort of a building's occupants[7]. Air quality can be portrayed by an Air Quality Index (AQI), a standardised scoring system to measure air quality. However, various indexes are compiled by different organisations and countries, which all vary. The UK's most commonly used index is the Daily Air Quality Index, specified by the Committee on Medical Effects of Air Pollutants (COMEAP). This system is a banding structure where the overall index is determined by the highest value of the index obtained based on the individual gases [8]. The AQI used in the USA was developed by the United States Environmental Protection Agency (EPA). This version of the AQI is defined by a piecewise linear function of the pollutant concentration [8].

These standards are very effective ways to visualise and understand the severity of air quality outdoors. However, in the indoors, the pollutants we look at differ slightly. Pollutants such as Sulphur Dioxide (SO_2) and nitrogen Dioxide (NO_2) are less prevalent indoors, while other contaminants, such as many Volatile Organic Compounds (VOCs), usually occur in higher concentrations indoors than outdoors [9]. As such, these AQI standards are less suitable for measuring indoor air quality; many existing IAQ solutions use a proprietary index to visualise IAQ. These indexes are very effective in terms of real-time visualisation of air quality. Still, from an analysis point of view, it is more effective to look at the air quality by breaking it down into its individual air quality factors/pollutants.

1.1.1 Existing IAQ solutions

Figure 1.1 Standalone IAQ Monitoring Device

There are many existing devices and products that are capable of monitoring IAQ and improving it. [10] The most common are probably commercially available Standalone IAQ sensors, as shown in Figure 1.1. These devices are usually internet-connected devices that monitor the level of some gases that affect IAQ and inform the user through multiple possible means. Some of these devices also perform some simple analysis of the data obtained. However, these devices can only monitor and report and cannot change or improve the IAQ in buildings, so they are often used in conjunction with air filters, etc. We also have to consider that due to the nature of these devices, the sensors used are not the most accurate[11].

Figure 1.2 Portable IAQ Monitoring Devices

Another known IAQ Solution is Handheld sensors, as shown in Figure 1.2. These sensors are very accurate but expensive, and they are used to measure the gas level of different IAQ factors one moment at a time and not over long periods [12]. These handheld sensors are very effective for some IAQ factors, such as radon, but not as useful for PM and VOC. This is because some air pollutants, if present, such as radon, do not vary much over time. However, some other factors, such as Volatile Organic Compounds (VOCs) and Particulate Matter (PM), vary significantly depending on what is happening in the surroundings and

within the building. Therefore, we should consider using these handheld solutions in conjunction with other solutions [13], [14].

Figure 1.3 BMS Attached IAQ Device

Figure 1.3 shows IAQ sensors attached to the Building Management System (BMS) of a large building, which is an existing solution in relatively newer buildings[15]. This solution can measure IAQ factors and immediately attempt to resolve the issue, such as turning the ventilation up. These systems are usually designed to provide immediate reactions to the situation within the building and do not usually consider past data. These solutions also need to be integrated into the infrastructure of the building and require custom programming and designing. They usually incur extra costs to implement, resulting in them only being practical in newly built offices and large buildings but not smaller buildings and households.

The existing commercial and applied solutions show that existing devices and systems only show real-time or historic sensor results, and no prediction or advice on future IAQ is provided. This results in any action based on these solutions being reactive towards observed air quality changes. Therefore, this solution, which uses neural networks to obtain predicted accurate IAQ results, is necessary to make proactive changes to the air quality to prevent spikes in air pollutants instead of mitigating them after they happen.

1.2 Novelty

1.2.1 Novelty 1 – Multisite model

The primary innovation of the present study lies in implementing a multi-site model that links predictions between sites. This was actualised by integrating a shared hidden state among multiple Long Short-Term Memory (LSTM) models. The initiative addresses several site-specific attributes associated with applying LSTM to indoor air quality (IAQ), with a potential extension to analogous applications. Traditionally, LSTM methodologies in IAQ necessitate site-specific data for model training prior to any predictive endeavours. This convention requires a preliminary training phase to enable the model to generate predictions.

The employment of the multi-site model yielded two discernible advancements when juxtaposed with the conventional single-site methodology. The first advancement is the extension of the forecast horizon, which implies an enhanced capability of the model to project further into the future with augmented accuracy.

The second advancement pertains to reducing the minimum training duration for the model, contingent upon specific conditions. In scenarios where both sites are subjected to concurrent model training, no conspicuous variance in training duration is observed. Conversely, when one site undergoes initial training with the subsequent inclusion of a second site at a later juncture, a notable diminution in the minimum training duration for the latter site was discerned.

1.2.2 Novelty 2 – Expanding Training method.

Another distinctive aspect of this study is incorporating a form of error correction into the LSTM model. We investigated three training methodologies: the conventional approach entailing a fixed training duration, a shifting training paradigm wherein the training period transitions in tandem with the predictions, and lastly, an expanding training approach. In the expanding training method, the model is initially trained over a predetermined duration, followed by introducing an error correction phase extending beyond this juncture. Through this error correction phase, the model's training process is iteratively refined, thereby potentially enhancing its predictive accuracy and adaptability across varying datasets and temporal frameworks.

2 Background

This chapter commences by examining the individual air quality factors considered in our study and their implications on human health. Additionally, we explore the sensors designated for measuring each respective air quality element. Subsequently, we delve into the assorted wireless technologies evaluated during the design phase of the proposed indoor air quality monitoring system. The discourse then explores the theoretical reasoning for employing Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks, in air quality predictions. The utilisation of LSTM, in particular, underscores its aptitude for deciphering time-series data, which is essential for accurate forecasting in indoor air quality.

2.1 IAQ Factors and Respective Sensors

Factors that affect the Indoor Air Quality amongst those factors regularly looked at are Particulate Matter (PM), Volatile Organic Compounds (VOC), Radon, Carbon Monoxide, Carbon Dioxide, mould, and Nitrogen Dioxide[16], [17]. Contrary to what many people may think, the origin of some of these compounds is not limited to the outdoors; for example, PM can also be emitted from smoke during cooking. Furthermore, pollutants such as VOCs are primarily indoor pollutants often emitted from furniture, carpets, paints, etc., within the property.

The factors we choose to look at initially are VOCs, Carbon Dioxide, Temperature, Humidity and Particulate Matter (PM), specifically PM 2.5. We also considered temperature and humidity as they have possible effects on other IAQ factors, such as mould, and helped us compensate for errors in the eCO_2 reading and TVOC readings.

2.1.1 Particulate Matter (PM)

We specifically look at PM 2.5 (particulate matter of diameter smaller than 2.5 µm) as these particles are considered hazardous compared to larger particles. They can penetrate deep into the respiratory system and the lungs, as they can pass through the filtration of nose hair[18]. PM2.5 has a significant adverse effect on the human respiratory system, and about 5% of all deaths are estimated to be related to PM2.5. In the UK, that is 30,000 yearly deaths [19], [20], [21], [22], [23], [24], [25]. Furthermore, with the COVID-19 situation, there is potential evidence that the risk of death due to COVID-19 correlates with exposure to high PM levels [26]. The unit of measure for PM 2.5 is micrograms per cubic meter of air (µg/m3), and according to the UK standard, healthy levels are below 16 µg/m3[27]

Sensor

Particulate Matter sensors work using a light detector and a beam of light. The Sensor sits at the angle to the beam of light, and as particulates pass the beam of

light, some light is reflected onto the sensor. As a fan is used to move air at a steady rate through the beam of light, the length of the pulses and quantity of pulses, the size and concentration of any particulate matter in the air can be found. [28]

2.1.2 Total Volatile Organic Compounds (TVOC)

Volatile Organic Compounds (VOCs) are another important group of air pollutants known to contribute to many serious health-related impacts. They have been linked to symptoms such as irritations of the nose, throat, and eyes, causing headaches, nausea, dizziness, and allergic skin reactions. They can also damage the internal organs such as the liver and kidneys. Moreover, some compounds of VOCs, such as Toluene and xylene, may not be immediate hazards but can lead to chronic health risks, which could result in serious neurosis[29], [30], [31]. Due to the large variety of VOCs, we use a Total VOC (TVOC) sensor that looks at the total concentration of multiple airborne VOCs. The unit of measurer for TVOC is micrograms per cubic meter of air (µg/m3), and according to the UK standard, healthy levels are below 300 µg/m3[27]

Sensor

The Total VOC sensors looked at are called metal oxide (Moxa) sensors. These sensors work by heating a thin film, or surface, of the metal-oxide nanoparticle to about 300℃. The film will adsorb oxygen particles onto the surface. These oxygen particles will react with the VOCs in the air, resulting in the release of electrons from the oxygen and thus affecting the electrical resistance of the Metal Oxide Layer. This resistance can then be measured; therefore, we get a reading of TVOC values.[32]

2.1.3 Carbon Dioxide (CO_2) & Estimated Carbon Dioxide (eCO_2)

Carbon Dioxide is another major gas considered when looking at indoor air pollution. Exposure to increasing CO_2 is known to cause decreased concentration and drowsiness, and prolonged exposure has also been linked to changes in bone calcium and negative effects on the body's metabolism.[33], [34] eCO_2 is an estimator of current CO_2 concentration by rescaling some easier-to-measure quantities such as TVOCs and Hydrogen Gas. The unit of measurer for eCO_2 is parts per million(ppm), and according to the UK standard, healthy levels are close to 00 ppm and below 800 ppm.[27]

Sensor

When looking at CO_2 sensors, we looked at both actual CO_2 sensors as well as eCO_2 sensors. In terms of actual CO_2 sensors, the most common type is the Non-Dispersive Infrared (NDIR) CO2 sensor. These sensors work in the principle that each atom and molecule can absorb light of a specific frequency. As such, these sensors work by

shining a light on the specific frequency for CO_2 in a small, closed chamber and measuring the amount of light that reaches the other end of the small chamber. By doing so, different amounts of CO_2 result in a different amount of light being absorbed, and we can obtain the CO_2 concentration.[35]

On the other hand, we also look at eCo2 Sensors. These sensors are Metal Oxide sensors, the same as TVOC sensors, where the resistivity of the sensor changes depending on CO_2 concentration. We chose to use eCO_2 Sensor due to the significantly lower cost of eCO_2 Sensors compared to actual CO_2 sensors [32]

2.1.4 Temperature and Relative Humidity

Extended exposure to low Indoor air humidity has been shown to influence perceived IAQ, sensory irritation symptoms in eyes and airways, work performance, sleep quality, virus survival, and voice disruption. As absolute humidity requires large sensors to measure, we choose to take measurements of temperature and relative humidity, which can be used to obtain the humidity values of a space. As such, temperature and relative humidity are factors that should be considered when looking at IAQ [36], [37].

Temperature Sensor

There exist various types of temperature sensors, the most common of which are thermistors, thermocouples and semiconductor junction sensors.

Thermistors are devices whose resistance changes with temperature. Thermistors are passive resistive devices, which means we need to pass a current through it to produce a measurable voltage output.

Thermocouples are by far the most common type of temperature sensor due to their simplicity. Thermocouples are thermoelectric sensors that basically consist of two junctions of dissimilar metals, such as copper and constantan, that are welded or crimped together. One junction is kept at a constant temperature, called the reference (Cold) junction, while the other is the measuring (Hot) junction. When the two junctions are at different temperatures, a voltage is developed across the junction, which is used to measure the temperature.

Lastly, semiconductor junction temperature sensors work by monitoring the characteristics of a transistor within the integrated circuit or outside it. Transistors have slightly different properties at different temperatures, and as such, the sensor will monitor these properties to gain an accurate value of the temperature of the said transistor and, thus, an excellent estimate of what the ambient temperature is. [38]

We chose to use semiconductor junction temperature sensors due to their small size and the availability of a semiconductor junction temperature sensor with integrated humidity sensors.

Humidity Sensor

Semiconductor humidity sensors work by placing a thin strip of metal oxide between two electrodes; the capacitance between the electrodes then changes at different relative humidity as the electrical capacity of the metal oxide is affected by the relative humidity [39].

2.2 Wireless Technologies

When designing the proposed indoor air quality monitoring system, multiple communication protocols were considered to link sensors. For flexibility in sensor placement, we decided to use a wireless communication method. The following wireless protocols were compared and considered for this application.

Technology	Power Consumption	Bandwidth	Range	Requires infrastructure	Indoor penetrative power
LTE-M	Medium	1Mbps	10km	No	High
WIFI	Medium	288.8Mbps	100m	Yes	Med
Zigbee	Low	100kbps	50m(Mesh)	No (Mesh)	Low
NB-IOT	Low	200kbps	10km	No	High
LoRa	Low	50kbps	20km	Yes	High
Bluetooth	Low	2Mbps	100m	Yes	Low

Table 2.1 Comparison of different wireless technologies.

Table 2.1 shows multiple wireless technologies that were considered during the design of the air quality data acquisition system. [40], [41] Each of the technologies had both benefits and some disadvantages. LTE-M and NB-IOT were both considered because we would not need to set up an infrastructure for gateways. WIFI was also considered because many sites would already have an existing WIFI infrastructure. Bluetooth Low Energy has a very low energy consumption while keeping a decent bandwidth but would have needed multiple gateways at the site due to its limited range. Mesh networks such as Zigbee are ideal in a situation where we have high-density data acquisition units, but where the units are less dense and more spread apart, we could run into issues. Finally, LoRa, which has a very long range while keeping a low energy consumption, would work well in a situation where modules are both close to the gateway or very far from the gateway. LoRa, in general, is a rising protocol in the IoT area. As such, we considered both LoRaWAN and LoRa using custom Gateways. LoRaWAN would result in simple infrastructure in places like Amsterdam, which has a city-wide LoRaWAN network. In the case of just using LoRa and custom gateways, we would have to make gateways for the modules to connect to, but this would work in this situation as these gateways would be an ideal location to store an information

database. Furthermore, using LoRa and custom gateways would significantly reduce energy consumption compared to LoRaWAN.

2.3 Neural Networks for Time Series Predictions

2.3.1 Recurrent Neural Networks

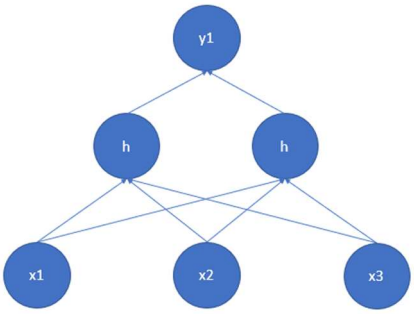

Figure 2.1 TNN Neural Structure

A traditional Neural Network (TNN) takes a fixed-size vector input. This limits the usage of a conventional neural network to a situation which involves a series of inputs with a fixed, predetermined size. Figure 2.1 shows a traditional neural network with an input of size 3(x1, x2, x3), a hidden layer of size two and an output layer of size 1(y1)

A TNN would have limited functionality in applications where we are looking at a situation that involves series inputs with no predetermined size. We could call a TNN multiple times for each input (x1, x2, x3) in the series to compute each output(y1,y2,y3). However, this would result in each of the networks not considering that one of the inputs may affect the others and would result in multiple single input single output Neural Networks, as shown in Figure 2.2.

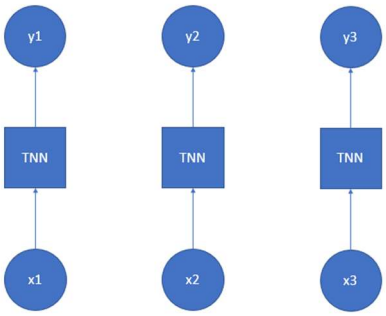

Figure 2.2 Multiple Input TNN

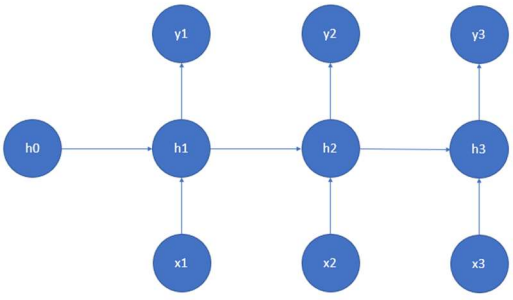

Figure 2.3 RNN Neural Structure

As such, we bring forth The Recurrent Neural Network, shown in Figure 2.3. This type of neural network remembers the past not only during training but also things they learned during prior inputs while generating outputs. Therefore, they can have one or more input vectors and produce the same number of output vectors where the outputs are not only affected by their respective input but also a hidden state vector which represents the prior learnt information. [42], [43], [44]

2.4 LSTM Theory

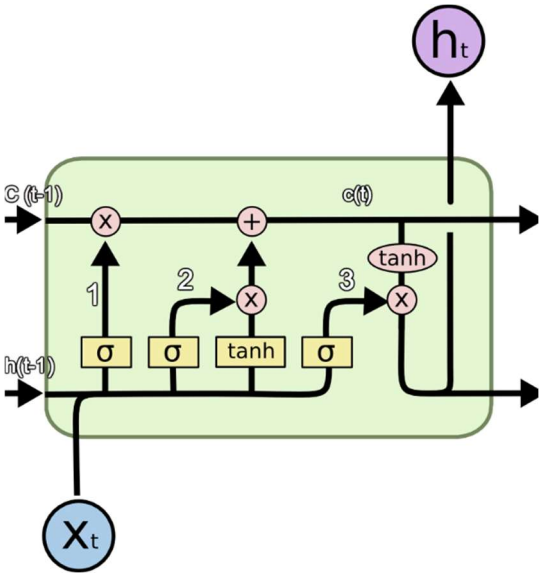

Figure 2.4 LSTM Cell Structure

LSTM is a special kind of recurrent neural network (RNN) that focuses on resolving issues most RNNs have with long-term memory. In the average RNN, every time a new set of inputs enters the Neural Network, the network's "memory" grows bigger and bigger. Over time, this results in an unstable network due to the accumulation of error gradients during updates. LSTMs, in the other case, are designed such that retaining information for prolonged time periods is the default setting. This is achieved through the incorporation of the LSTM gates. A typical LSTM cell has three gates: forget, input, and output. In Figure 2.4, these are depicted as the three sigmoid layers.

LSTM Forget Gate: $$f_t = \sigma_g(W_f x_t + V_f h_{t-1} + b_f) \quad (2.1)$$

LSTM Input Gate: $$i_t = \sigma_g(W_{i^x t} + V_i h_{t-1} + b_i) \quad (2.2)$$

LSTM Output Gate: $$O_t = \sigma_g(W_o x_t + V_o h_{t-1} + b_o) \quad (2.3)$$

LSTM Cell Input: $$\tilde{c}_t = \sigma_c(W_c x_t + V_c h_{t-1} + b_c) \quad (2.4)$$

LSTM Cell State: $$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad (2.5)$$

LSTM Hidden State: $$h_t = O_t \circ \sigma_c(c_t) \quad (2.6)$$

Equation 2.1- Equation 2.6 are the LSTM equations which the notations can be described as follows.

- **W_f, W_i, W_C, W_o:** Weight matrices w.r.t gates and cell state
- **b_f, b_i, b_C, b_o**: Biases w.r.t gates and cell state
- **σ**: Sigmoid Activation function outputs a value between 0 and 1 for any given input.
- **tanh**: Tanh Activation function outputs a value between -1 and 1 for any given input and has a steeper gradient as compared to sigmoid.

Breaking down the LSTM cell in Figure 2.4, we can understand how the cell functions. The operation of each cell can be broken down to each of its gates.[42], [43], [44]

2.4.1 Forget Gate – 1

The white number 1 in Figure 2.4 shows the forget gate, while the equation which is responsible for deciding what part of the cell state from the previous timestep (C_{t-1}) must be forgotten. The sigmoid activation is used to output values between 0 and 1, where 1 represents "completely keep this" while 0 represents "completely get rid of this". Equation 2.1 shows the mathematical equation for the sigmoid function of the forget gate.[45]

2.4.2 Input Gate – 2

The white number 2 in Figure 2.4 shows the input gate(i_t) responsible for determining if information should be saved to the cell state or left behind.

Now that the data to be removed has been handled by the forget gate, we need to evaluate what data must be carried to the next time step. This is done in two parts. The first part involves the sigmoid function of the input gate (i_t), which is described by Equation 2.2. It determines what data carried by the cell state must be updated and carried forward to the next time step.

The second part is a tanh layer that creates a vector of new values ($\tilde{C}_t$) that can be added to the current cell state, which is described by Equation 2.4. Tanh activation pushes the values between -1 and 1 and inhibits the data that we do not wish to add to the cell state.

We can now use Equation 2.5 to decide the information to carry to the next timestep (C_t) from the outputs of the input gate, new values added to the cell state, forget gate and the cell state from the previous timestep.[45]

2.4.3 Output Gate – 3

The white number 3 in Figure 2.4 shows the output gate (o_t). This output gate, in combination with the current cell state C_t, obtained earlier, is used to determine the output at each timestep.

The output gate, which is also a sigmoid layer shown in Equation 2.3, decides which parts of the cell state we wish to output. Finally, we put the cell state through tanh described by Equation 2.6 and multiply it by the output of the sigmoid function to determine the hidden state for the next LSTM cell (h_t) [45]

2.5 Performance Indicators

To evaluate the performance of prediction models for indoor air quality (IAQ), prediction accuracy is measured by comparing the predicted values to the actual data. The discrepancy between these values at any given moment represents the error at that point in time. In this thesis, error values are frequently plotted over time to provide a visual representation of the model's accuracy.

For a numerical summary of accuracy over the testing period, the Root Mean Square Error (RMSE) is utilized. Additionally, the Percentage Root Mean Square Error (%RMSE) is employed to compare error rates across variables with differing scales. The %RMSE normalises these differences into percentages, making it a valuable metric for cross-variable comparisons. In some studies, %RMSE is also referred to as the normalized root mean square error.

Some further performance indicators, include the computational time and the training parameters which include training duration, prediction duration and Training generations.

2.5.1 Root Mean Square Error & Percentage Root Mean Square Error

To measure the accuracy of the model, we compare the prediction error from the actual data. To measure this over a period of time, we use the root mean square of this error and the percentage root mean square of this error, which we will refer to as RMSE and %RMSE, respectively. This is calculated using the following equations.

RMSE Equation：

$$RMSE = \sqrt{\frac{1}{n}\Sigma\left(Y_p - Y_T\right)^2} \tag{2.7}$$

Where

n = number of non-missing data points

Y_p = predicted time series

Y_T = actual observations time series

%RMSE Equation：

$$\%RMSE = \frac{\text{RMSE}}{Y_{max} - Y_{min}} \tag{2.8}$$

Where

Y_{min} = minimum value of time series

Y_{max} = maximum value of time series

A smaller %RMSE value indicates higher prediction accuracy. However, when comparing %RMSE values in IAQ studies, it is important to consider that many factors beyond the model itself—such as training parameters—can influence %RMSE outcomes. Models trained more extensively generally outperform less rigorously trained models, though at the cost of increased computational demands.

2.5.2 Computational time/complexity

Computational time refers to the duration required for the model to be trained, validated, and tested on a computer. In this thesis, all tests related to computational complexity were conducted on the same device to ensure consistency and enable relative comparisons.

Regarding computational specifications, the experiments were performed on a system using a single-GPU configuration equivalent to an NVIDIA RTX 3090, capable of approximately 16 tera floating-point operations per second (TFLOPS). This setup provided sufficient computational power to evaluate the models while maintaining consistency across experiments.

2.5.3 Training Duration

Training durations refers to the amount of data used to perform the initial training of the model before the model makes predictions. Units for this performance indicators will generally be; days, weeks and months.

2.5.4 Prediction Duration

Prediction duration refers to the time horizon for which the model generates predictions, such as forecasting 1 hour or 3 hours into the future. This performance indicator is typically measured in units of minutes or hours, depending on the scope of the prediction task.

2.5.5 Training Generations

Training generations refers to the number of times the model processes the training data before it attempts to make predictions. This performance indicator is typically in the scales of tens and hundreds of generations.

2.5.6 Performance indicator Caveats

It is however to note that variations in computational configurations, datasets as well other hyperparameters within LSTM across studies can result in discrepancies, even when attempting to replicate another study's model. To address these challenges, this thesis compares results against the base LSTM model and models from other studies, ensuring all comparisons use our dataset, computational equipment, and fixed training parameters to maintain consistency.

3 Literature Review

This chapter will be initiated by examining alternative analysis models tailored for indoor air quality and air quality in general, which have deviated from the Neural Networks approach. Following that, the chapter then looks at neural network analysis approaches primarily LSTM and GRU to look at indoor air quality and their pros and cons. The chapter then looks into a couple of multisite LSTM based models which were proposed for predicting air quality.

3.1 Alternate Analysis Methods

Various alternate techniques have been proposed for IAQ. They include Multilayer feed-forward, Multi-level temporal regression, support vector machines, and autoregressive models. We will discuss them in this section.

3.1.1 Multi-level Temporal Regression Models

Multi-level temporal regression models have been used extensively to predict air quality. These models leverage spatial-temporal covariance functions, allowing them to model data dependencies over space and time. [46]. Some of these models with complete spatial-temporal covariance functions have achieved very accurate predictions but require a high computational cost to achieve this. [46]. Increasing the complexity of the covariance function while increasing the complexity of its hierarchical structure has resulted in lower but acceptable accuracy at a reduced computational cost. These models only apply to predictions based on data points from fixed location datasets and cannot be applied to new air quality monitoring sites.

The balance between model complexity and computational cost is a recurring theme in air quality prediction. While these models are powerful in specific scenarios, their inability to generalise to new sites limits their scalability and utility for broader applications, such as nationwide air quality monitoring systems.

3.1.2 Support Vector Machines

SVMs have been applied to predict future pollutant levels and have the advantage of being computationally efficient compared to more complex models like temporal regression models. However, this comes at the expense of accuracy, typically achieving about 70-80%. SVMs are best suited for scenarios where the trade-off between computational cost and accuracy is acceptable and fast results are required. [47]

SVMs represent a compromise in air quality modelling. They offer lower computational requirements but at the cost of accuracy. This study showed that a suitably configured SVM can achieve %RMSE of 20-30% while keeping with low computational requirements. This trade-off makes them suitable for quick estimations but less effective when high precision is necessary.

3.1.3 Multivariate analysis of variance (MANOVA)

Multivariate analysis of variance (MANOVA) is a statistical method used when looking at indoor air quality or air quality in general. MANOVA models, once built, provide an accurate way to predict air quality based on related factors and have offered significant success in predicting PM values in the indoor environment. However, a constructed model from one site does not always apply to other sites, as the conditions. [48], [49]Although accurate, this model requires building on a site-to-site basis and proves relatively difficult to automate. The major negative aspect of applying this method is its lack of scalability and automobility, but at the same time, it allows for a deeper understanding of why and what causes the relationship between air quality factors and the variables that affect them.

MANOVA excels at capturing complex relationships in air quality data but lacks the scalability necessary for widespread deployment. Its strength lies in its ability to identify causal factors, but the model's site-specific nature limits its general applicability.

3.1.4 Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average (ARIMA) is one of the most popular statistical methods for time series analysis. As such, there has been some success in using this method to look at indoor air quality prediction [50]ARIMA can be divided into two categories: the ARIMA and the seasonal ARIMA, called SARIMA, used when there is a periodicity in the data series instead. ARIMA predictions evolve over time, using recent data close to the predicted period following the process changes as input. Therefore, the ARIMA models adapt quickly to possible variations of the series, but they pay this quality in terms of short forecast periods. [50]. An ARIMA model has 3 primary components that need to be calibrated in order for the model to be used effectively. These components usually have varying values for different sites as well as different forecast periods. This results in some difficulty in automating this model as these components would need to be recalibrated for not only different sites but also different configurations within the same site. Once calibrated this study has shown that ARIMA can achieve prediction accuracies of over 90%.

ARIMA's strength lies in its ability to adapt to time-based changes in data, making it suitable for real-time air quality predictions. However, the complexity of parameter tuning limits its scalability and automation, especially in dynamic environments where air quality data varies across regions.

3.1.5 Hierarchical agglomerative cluster/Multilayer Feed Forward

Hierarchical agglomerative cluster analysis has provided a method to identify major sources of indoor air pollutants. Using this method, 18 variables that largely influence indoor air quality were determined. Principal component analysis of each cluster revealed that the main factors influencing the high complaint group were fungal-related problems, indoor chemical dispersion, detergent, renovation, thermal comfort, fresh air intake location ventilation, air filters, and smoking-related activities. [51].

Multilayer feed-forward neural networks have also been used to identify and categorise sources of pollutants in the indoor environment. [52]. Both methods have had great success in identifying indoor air quality sources in the case of hierarchical agglomerative cluster analysis. This has allowed for identifying generic sources of pollutants but not specific factors in households that can be changed to improve air quality. At the same time, Multilayer feed-forward neural networks have allowed for this identification. However, this information is site-specific, and the neural network must be retrained at each site to provide reliable results.

Both methods excel at identifying sources of pollutants but suffer from the same limitation as other models—the need for retraining for new sites. This restricts their use in large-scale or widely distributed systems but makes them valuable for in-depth analysis of air quality in specific environments.

3.2 Time Series Data Analysis – LSTM/GRU

Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have provided the most success among all time-based machine learning methods in predicting indoor air quality. Both LSTM and GRU have provided similar success in predicting indoor air quality. As GRU reduces the number of gates and essential parameters in its analysis, there is some debate on whether it loses accuracy by doing this. [53]. As with the other analysis methods, LSTM and GRU are also site-specific models, and trained models cannot be relied on when applied to another site, but the advantage lies in the ease of automation that these methods offer. As such, LSTM and GRU can enable the creation of an automated system that can analyse indoor air quality and its factors without much interaction.

The primary advantage of LSTM and GRU is their ability to handle complex time series data while remaining relatively easy to automate. However, the site-specific nature of these models poses a challenge for widespread application.

3.2.1 Internet of Things (IoT) Based Indoor Air Quality Sensing and Predictive Analytic

This study looked at the deployment of Wi-Fi-based low-cost air quality sensors that collect data and perform analysis on the cloud. [54]. This study used an LSTM model to forecast the upcoming air quality in the deployed locations. Their model achieved an accuracy of 99% when provided with approximately 2 months of training data using a prediction duration of 1 hour. This approach, however, is site-specific, like most other LSTM deployments and will need to be retrained for every site.

This method demonstrates the potential of IoT in improving indoor air quality monitoring with promising results. However, the reliance on LSTM's site-specific retraining poses a scalability challenge, especially in environments where conditions differ significantly from the original training data.

3.2.2 IndoAirSense

This study looked at a proposed framework called IndoAirSense. This approach deployed sensors in specific university classrooms. They first used multilayer perceptron (MLP) and eXtream Gradient Boosting Regression (XGBR) to estimate the real-time IAQ of the other classrooms without sensors. Following that, they used LSTM-wf, a modified Long Short Term Memory (LSTM) without the forget gate, to make predictions of the upcoming air quality. Removing the forget gate improved the training time as the LSTM model is considerably less complex while maintaining an overall %. However, removing the forget gate, which keeps the long-term memory in LSTM, resulted in the model being unable to detect and forecast the anomalies and sudden random spikes in the data. [55]. The prime benefit of this approach seems to be the incorporation of MLP and XGBR, which provided very accurate estimations of the IAQ in adjacent classrooms without sensors. It is likely that the estimation accuracy is due to the fact that these classrooms probably had similar physical characteristics. Will this accuracy persist if the estimation is made of a classroom in a different location or a room with different characteristics that have yet to be tested?

The IndoAirSense framework is innovative in its combination of MLP, XGBR, and LSTM, achieving high accuracy in un-sensored locations. However, the trade-off in anomaly detection highlights the risk of oversimplifying models to gain speed.

3.2.3 Combination GRU and LSTM

This study introduced a combined predictive approach that employed two variations of the recurrent neural network (RNN) model, specifically the gated recurrent unit (GRU) and long short-term memory (LSTM) models [56]. Their objective was to forecast the daily air quality index (AQI) for the major cities of Dhaka and Chattogram in Bangladesh. Their approach involved utilising GRU and LSTM as the initial and subsequent hidden layers, respectively. These were followed by two dense layers functioning as a prediction model. The outcomes demonstrated that their model accurately tracked the AQI patterns for both cities and highlighted the enhancement in overall performance achieved by employing both GRU and LSTM models, in contrast to using either model individually. However, even this combined model retains the characteristics of being site-specific and retraining required for every site.

3.2.4 Multivariate and multi-output indoor air quality prediction using bidirectional LSTM(BiLSTM)

This study looked at using a bi-directional variation to LSTM to predict individual pollutant levels. This study used BiLSTM which is a variation of LSTM that learns the input sequence both forward and backwards and concatenate both interpretations. This study used a dataset of 5 months and used 60% of the data form training and 40% for validation and testing. With a prediction duration(forecast) of 1 hour the study achieved an %RMSE of 3-6% across all features measured using BiLSTM when compared to LSTM which only achieved an %RMSE of 6-9% across all features. [57]

This method demonstrates a simple yet beneficial method to improve the prediction accuracy of an LSTM model when applied to air quality data.

3.2.5 LSTM-Autoencoder-Based Anomaly Detection for Indoor Air Quality

This study proposed an LSTM-AE-based hybrid deep-learning technique for detecting contextual anomalies in IAQ datasets. [58] The incorporation of the auto-encoder layer in this approach reduces the data dimension and allows for the computation of an optimal reconstruction error associated with each time sequence. This reconstruction error is used as a threshold to detect contextual anomalies that deviate from the normal pattern. This model achieved an accuracy of 99.5%. This seems to outperform another similar model, which reached accuracies of up to 99.27%, but this difference is possibly just due to the varying characteristics of the datasets. [58] The incorporation of the autoencoder layer improved the training time of the model by reducing the data dimension of the LSTM model.

LSTM-Autoencoder models are effective in anomaly detection and improve training efficiency by reducing data dimensionality. However, the accuracy of anomaly detection may vary based on the characteristics of the dataset.

3.2.6 ARIMA-LSTM combination model optimised by dung beetle optimiser.

This study looks at a combination model of ARIMA and LSTM while using the dung beetle algorithm to optimise the LSTM's hyperparameters. This approach uses ARIMA to break the raw data up into linear and non-linear components. ARIMA is then used to make predictions on the linear components of the data, while LSTM is used to make predictions on the non-linear components. Here, the dung beetle optimiser is used to optimise the hyperparameters of the LSTM neural networks for each site or set of data input. [59] This approach used the normalised AQI instead of looking at the various air quality factors. One of the noticeable benefits of this approach was the reduced training time. Another significant benefit is the efficiency of the dung beetle optimiser, which reduces the time taken to optimise the model and keeps the model optimised efficiently. However, this model, similarly to the other LSTM models, is site-specific and requires retraining when datasets from different places are used.

3.2.7 Data-driven model for predicting indoor air in naturally ventilated educational buildings.

This study investigated the combined use of multiple machine learning (ML) techniques to enhance air quality in naturally ventilated schools, with a primary goal of identifying the key factors influencing indoor air quality in these settings. The methods focused mainly on a combination of multilayer perceptron, support vector machines (SVM), and long short-term memory (LSTM) networks. [60]Some success was achieved in analysing multisite data, particularly through the use of multilayer perceptron and SVMs. However, LSTM models encountered difficulties when applied across multiple buildings. The study achieved a mean test accuracy ranging from %RMSE values of 46.4% to 19.5%, with maximum test accuracies between 19.3% and 18%.

As the study progressed, efforts centred on using these techniques to identify factors that could improve indoor air quality on a one-time basis, rather than creating a dynamic model capable of adapting to changing conditions in real time.

3.2.8 Sequential prediction health risk assessment for the fine particulate matter using deep recurrent neural networks.

This study used ML techniques to find and improve air quality in the subway. The study concluded that LSTM and GRU were the most suited ML techniques to forecast air quality in indoor environments in general.[61], [62]. This study looked at incorporating what they called surrogate indicators into the model to help indicate when the air quality would deteriorate. These surrogate indicators included current airflow, time of day and number of people in the station. Using these surrogate indicators they successfully built a model that could successfully predict when the air quality would deteriorate based on the surrogate indicators. It was found that indoor environments such as each subway station tend to be microenvironments where characteristics of 1 environment's characteristics would vary.

The study proposed that to achieve sustainable IAQ monitoring, multiple GRU models for each subway microenvironment would need to be incorporated through the use of low-cost sensors, as the model for each environment and station would need to be trained separately.

3.3 Multisite studies

3.3.1 Multi-site and multi-hour air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering.

This study looked at using convolution neural networks (CNN), LSTM, a combination of CNN and LSTM(CNN-LSTM), as well as spatiotemporal clustering to predict air quality in both indoor and outdoor environments across multiple sites in Beijing. [63]This study chooses to use an approach to predict and measure the air quality index (AQI) instead of the individual air pollutant levels. In this study, it was found that using purely LSTM for multi-site predictions wasn't suitable due to the site-specific nature of the air quality data. However, using the CNN-LSTM combination, they managed to circumvent this issue and successfully used it to make multi-site predictions.

The study utilised two years of data, with 70% allocated for training (approximately 16 months) and the remaining 30% for testing and validation. Prediction durations ranged from 1 hour to 6 hours, though the study provided limited details on the LSTM's remaining hyperparameters. Results showed that LSTM and CNN-LSTM were the most effective models for multi-hour air quality predictions, with CNN-LSTM outperforming LSTM by 2–3% in terms of RMSE for multi-site predictions. For shorter forecast periods (1–2 hours), the performance of both models was comparable. However, the CNN-LSTM demonstrated superior accuracy for longer forecast periods (3–6 hours). Based on the data provided by the study, we estimate the %RMSE of the model using CNN-LSTM to be about 4%

The study concluded that LSTM was the optimal model for Air quality prediction. The performance difference between the LSTM and CNN-LSTM was relatively small, but CNN-LSTM had a higher computation complexity.

3.3.2 Forecasting urban air pollution using multi-site spatiotemporal data fusion method (Geo-BiLSTM)

This study looked at creating a multisite model using LSTM and a data fusion method before feeding the data through a BiLSTM model. The study used Krigan Interpolation to transform the data of a target site and its eight neighbouring sites to be used as an input into a large Bi-LSTM model. This study focused on predicting PM2.5 and O3 levels only in the outdoor environment, and no testing was done using indoor data. The study made comparisons of this model with the aforementioned CNN multisite study as well as standard LSTM, GRU and BiLSTM models. [64]

The study used two years' worth of data to train the model, followed by 2 months of validation and testing. Prediction durations of 96h were used for testing in this study, as the study primarily focused on the outdoor environment. However, some data on shorter prediction durations could be extrapolated from the graphs. At the prediction duration of 96h, the Geo-BiLSTM model achieved an RMSE of 34.42 when compared to 76.11 of a

standard LSTM model. Based on the data on these graphs, these RMSE values translate roughly to a %RMSE of 17% for the Geo-BiLSTM model and 38% for the standard LSTM model. The study found that in this prediction duration of 96h, the Geo-BiLSTM model achieved the best results using their test data.

The results shown through the use of this Geo-BiLSTM model show that by incorporating the Krigan interpolation for data fusion into a BI-LSTM model, they have achieved an improvement in large-scale prediction of air quality information. However, the research done only looks at using data from adjacent sites in the model, as their focus was looking into spatial relationships between sites; it would be interesting to look into incorporating sites which are further apart to see how this would interact with this model.

3.4 Summary

In this chapter we had begun by reviewing multiple alternative methods for IAQ analysis. Delving into these studies has furnished insights into certain indoor air quality characteristics, which could help understand some of the findings observed in this investigation. These alternative methodologies serve as a comparative framework, enriching our understanding of the nuances involved in indoor air quality analysis and prediction. Through this comparative analysis, we found that apart from LSTM there are a few methods that are suitable for application on IAQ. Methods such as SVMs and ARIMA are feasible alternatives to make time-based predictions, while techniques such as MLFF provided a way for data classification as opposed to predictions.

In reviewing various LSTM approaches, it is evident that LSTM emerges as both a prevalent and effective methodology for predicting indoor air quality. Its efficacy is manifested through the high predictive accuracy exceeding 95%, as observed in multiple studies. Nonetheless, a recurrent characteristic associated with these LSTM approaches is the site-specific nature of this application domain. Such site-specificity necessitates retraining the LSTM model for each distinct site within an application. This study ventures into a multi-site model to find interlinking attributes among various sites.

We had then investigated a couple multisite variation to LSTM. These model which albeit were applied to primarily outdoor air quality provide methods in which LSTM could be modified to create a model that incorporated spatial data on top of temporal data into the LSTM model.

4 Hardware and Training Methods

This chapter looks at the overall hardware architecture of the system as well as the specifications of the various hardware components selected, including their models and accuracies. This hardware design provides a foundation for data to be collected and, based on that, how the training of the model is performed. We then look at the various training methods attempted and how they perform for this application. One of these methods highlights one of my work's novelties. In this chapter, we will compare how these three methods work as well as the variations they will provide in the accuracy of prediction and computational time.

4.1 Hardware Design

4.1.1 Design Overview

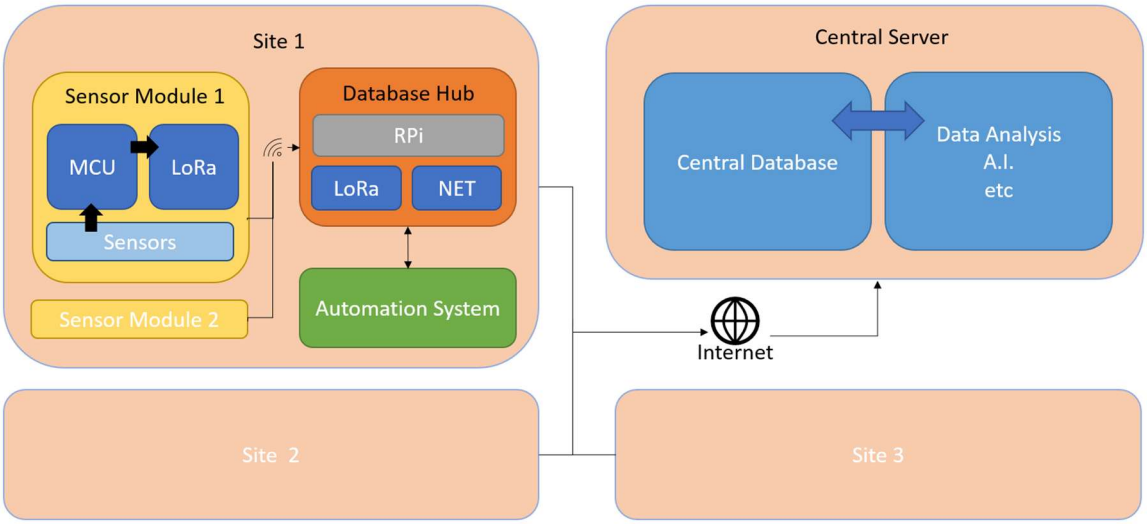

Figure 4.1 Data collection system architecture.

Figure 4.1 Highlights the design architecture of the data collection hardware. The system hardware comprises the sensor modules, database hubs and the central server. First, a series of wireless sensor IoT modules collect the air quality data every 1 minute. These sensor modules wirelessly transmit the data to the local database hubs using LoRa technology. The database hubs can also be integrated with the local home automation systems at the sites to collect additional data, such as the state of the lights as well as heating, ventilation and cooling (HVAC), at any moment in time. These database hubs then synchronise all the collected data with the centralised server over the internet.

One of the main aims of the design of this system was the scalability of the system. There are two scalable aspects of this system. First, each site's database hub is configured to allow up to 256 sensor modules. This limit is, by design, due to the allocated address size of 8 bits for each sensor node. The second scalable aspect is the number of sites linking to the central server. Our setup allows up to 150 sites to be connected to a single central server.

However, we are capable of increasing this number further through either some configuration changes or the use of intermediary servers.

4.1.2 Sensor Modules

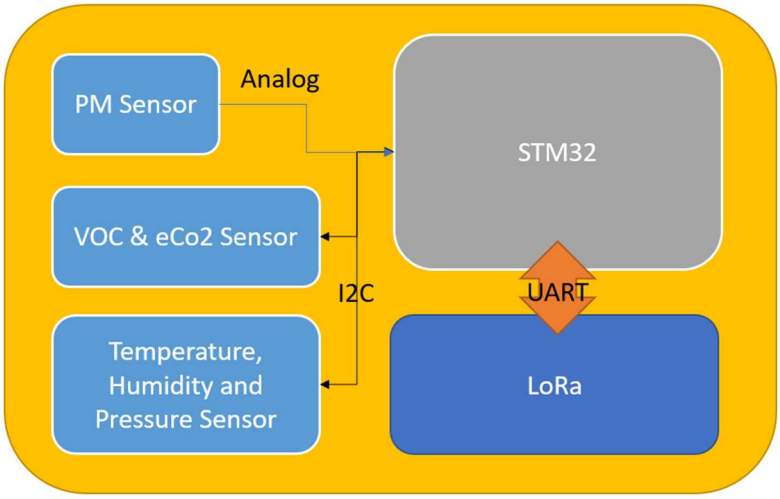

Figure 4.2 Sensor module architecture.

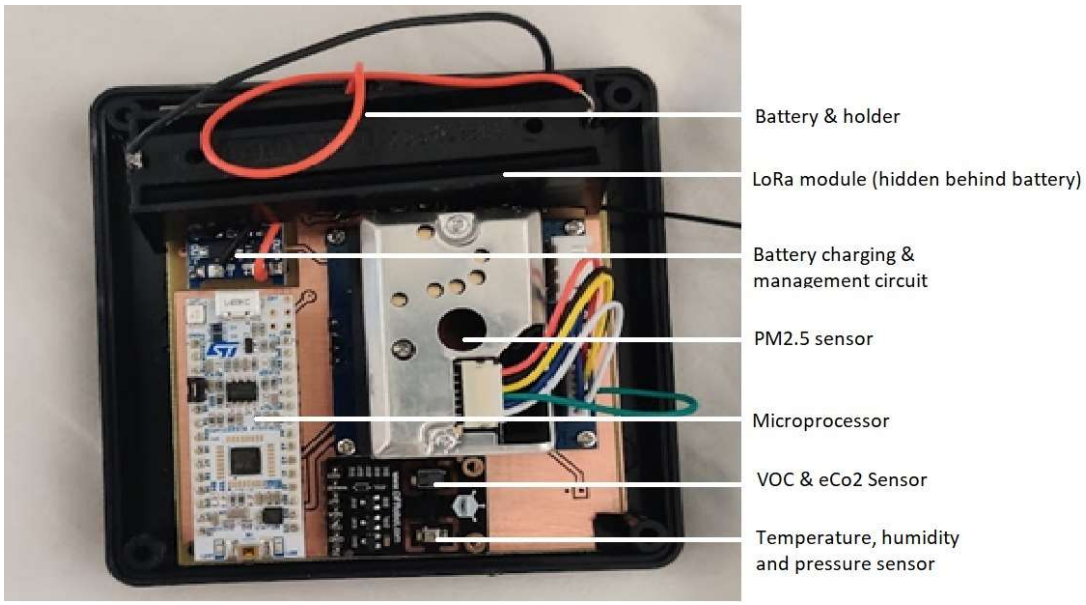

Figure 4.3 Sensor module.

Figure 4.2 shows the overall architecture of each sensor module, while Figure 4.3 is a physical picture of a sensor module without its cover showing all its components. These modules do not store data but send live data to the database hub every minute. A polling rate of once per minute was chosen as the sensor modules are mostly battery-powered, and we believed this to be a reasonable polling rate while still considering power consumption

and battery life. We will now look at the individual components of the sensor module and how they are connected to each other.

4.1.2.1 Main microprocessor

The microprocessor (MCU) used in the sensor modules is an STM32L432KCU6; specifically, we used the Nucleo-L432KC development board in these initial prototypes. We chose this microprocessor because it is a readily available ultra-low-power MCU. Some of the other specifications that affected the selection of this MCU were the 256Kb flash, 64Kb SRAM, built-in ADC, I2C bus, and SPI bus. [65], [66]

4.1.2.2 Wireless LoRa module

The wireless module used is a UCL in-house LoRa module. We selected this module for its low power consumption and max communication range of 11.2 km. This module communicates with the MCU using UART. When the MCU sends its strings, the module transmits them to the receiving LoRa module on the database hub.

4.1.2.3 PM 2.5 sensor

The PM 2.5 sensor used is a GP2Y1010AU0F. Some of the main factors in choosing this sensor were its low cost and the voltage required to power it. This sensor could be powered with anything from 2.5-5 V, and as such, we could power it directly from the built-in voltage regulator of the MCU, which supplies 3.3V. This sensor has an analog output of 0.5V/(100µg/m^3), which is fed into the ADC of the MCU. The sensor also has a sensitivity of 100µg/m^3. [67]

4.1.2.4 Temperature, humidity and pressure sensor

We used a BME280 as it is a combined temperature, humidity and pressure sensor which is readily available. Another primary reason for choosing this sensor was its supply voltage of 1.7-3.6 V, which the built-in linear voltage regulator of the MCU could directly supply. This sensor has a temperature accuracy of 1.25 °C, relative humidity accuracy of 3% and pressure accuracy of 100 Pa. This sensor communicates to the MCU via I2C. [68], [69]

4.1.2.5 TVOC and eCO_2 sensor

The TVOC and eCO_2 sensor used is a CCS811. This module was chosen primarily due to its low price to allow for mass deployment. However, it has a few drawbacks compared to other higher-priced TVOC modules. One of them is that the accuracy of the module is affected by the surrounding temperature and humidity and that the sensor needs to be heated up slightly to function. Fortunately, regarding heating, the sensor has a built-in heater to allow for this. Regarding the temperature and humidity affecting the accuracy, the manufacturer has supplied an algorithm that uses temperature and humidity readings to compensate for this variable accuracy. As we have live temperature and humidity values, we applied the algorithm to the MCU for all values pulled from the sensor. This sensor communicates to the MCU via I2C. [70], [71]

4.1.2.6 Battery & power

We used a TP4056 charging module and a Panasonic NCR18650B battery to supply power for the sensor module. The TP4056 allows us to power the module using the battery or directly with a 5V USB power supply. The module charges the said battery using the same

5V USB power input. The NCR18650B has a capacity of 3350 mAh, which can power the sensor module for approximately three months after a full charge. Power management is handled by the TP4056, which supplies 5 V to the MCU development board, which can then step it down using its built-in voltage regulators to 3.3V for the MCU itself, as well as the various other sensors and the wireless module.[72], [73]

4.1.3 Database Hub

The database hub consists of a Raspberry Pi Zero W (RPi) [74] Running RaspbianOS connected to a LoRa module developed in-house over UART. The LoRa module will send the data strings to the RPi upon receiving them from the sensor modules. The RPi will then parse the strings and store the individual sensor values, the sensor's identifier, and a timestamp in a MariaDB SQL database running on the hub. [75] We chose MariaDB due to its stable performance in data replication, which we used to connect the local database hubs to the central server. Using the collected data points, the hub can perform some simple analytics on data, such as running a trained model for short-term air quality predictions.

These database hubs also integrate with the existing automation system to collect more data and can control a building's automation. Integration is achieved through data communication protocols commonly used in building automation. The protocols include BACnet IP, RS232 through an in-house protocol, KNX, TCP socket via an in-house protocol, and Modbus.

4.1.4 Central Database

The final hardware component is the central database, a larger server that runs a MariaDB database. Our current application runs the central database on a Windows PC. The database hubs will replicate their databases through multi-master replication to this central database.[75], [76] Multi-master replication allows the central database to have a complete collection of all the data points of all the data collection sites cumulatively, while the database hubs only have the data points of their site. The central database can then perform more compute-heavy analytics on the data, such as multisite analysis and training of prediction models for the sites.

4.1.5 Sites and Sensor Distribution

For the initial testing and deployment of the system, we have deployed it at five sites with varying numbers of sensors at each site. Table 4.1 It shows the site locations and the sensor distribution at each site. The number of sites and locations were chosen and limited by the number of sites and locations for which we could get access to and consent for collecting data.

Site Reference	Location	Location Type	No. Of Sensors	Sensor Locations
Site A (1)	Islington	House	3	Lounge, Kitchen, Bedroom
Site B (2)	Stockwell	House	4	Lounge, Kitchen, Bedroom (2)
Site C	Chiswick	Office	2	Office space, Reception
Site D (3)	Euston	Apartment	3	Lounge, Kitchen, Bedroom
Site E (4)	Docklands	Apartment	2	Lounge, Kitchen

Table 4.1 Sites and sensor distribution

4.2 Training Methods

The hardware system described in 4.1 was crucial in collecting indoor air quality data across all study sites. LSTM models were selected for their demonstrated effectiveness in analysing air quality data, as elaborated in3.2. LSTM was intentionally chosen over GRU to maintain consistency in our testing framework. The analysis process using LSTM commenced with a thorough investigation of multiple training approaches, emphasising data input structures and prediction methodologies.

4.2.1 Description of Methods

When applying LSTM on a data set where the sample data constantly grows in size, we observe that the traditional way of using fixed duration in LSTM is not always ideal. As in some applications of LSTM, the accuracy of the predictions can decrease over time. This loss of accuracy is due to changing characteristics of the sample data, which is caused by the large number of unaccounted factors that can affect the data. Our indoor air quality dataset broadly fits into this type of dataset due to the large amount of human and environmental factors that can affect the data.

In order to circumvent this loss of accuracy over time, we will look at the three variations of how we have applied LSTM to the data set, including the aforementioned traditional method. To simplify the description of the methods, we assume we have a fixed sample of data N seconds long instead of an ever-expanding data set.

4.2.1.1.1 Fixed training duration

Figure 4.4 Fixed training method.

In the traditional method of applying LSTM, a fixed duration of data is used as the training data, which is run through the LSTM model repeatedly. This method is the traditional way of applying LSTM onto a dataset. We label this fixed training duration as T. We then obtain a fully trained model based on the data from the training duration. The trained model is then used to create a prediction for the rest of the sample data. We label this prediction duration as P. The prediction data is compared to the collected data, and the error between the predictions and the actual data measures the performance. Things to note with this method is that the total size of the data is always equal to the sum of T and P, as we can see in. Figure 4.4. The model also does not give any additional input past the training duration. In our testing and comparison of the training methods, we used a training duration of 4 weeks

(T=4 weeks). For the traditional method, we tested with a prediction duration of 1 month up to 4 months.

4.2.1.1.2 Shifting training duration (Moving method)

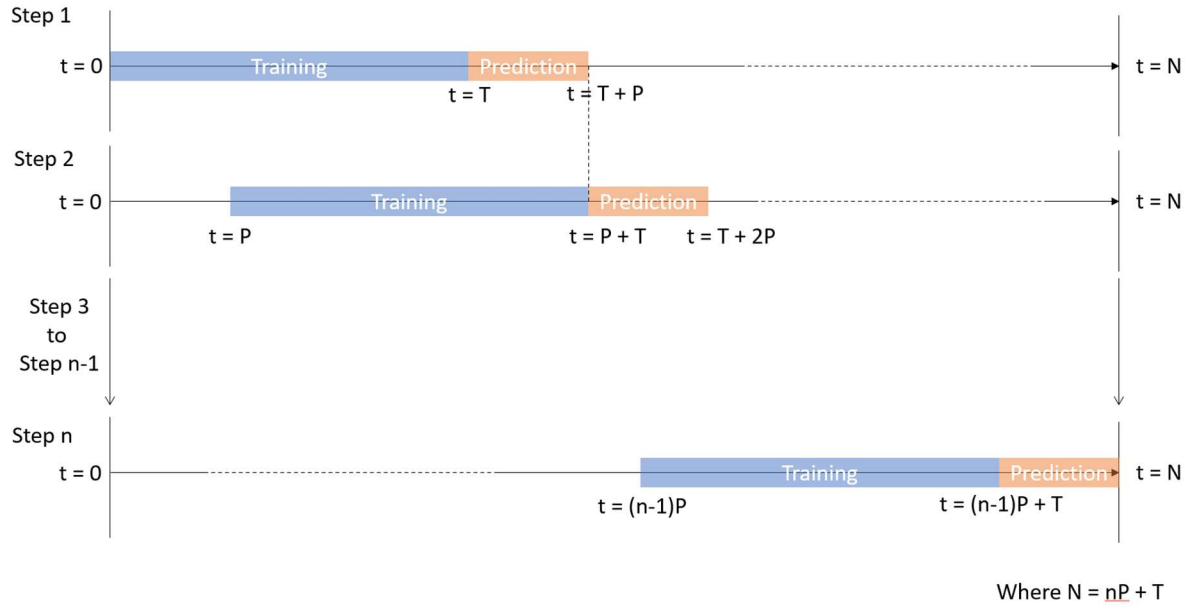

Figure 4.5 Shifting Method

Unlike the traditional method, the shifting training duration method is split into multiple steps, as shown in Figure 4.5. Using this method, we create shorter prediction durations, each of size P. As we can see in Figure 4.5, step 1, the initial training duration(T) is the same as the fixed method, but the prediction duration(P) is shorter and does not cover the whole sample duration. In Step 2, we shift the training and prediction sections by P away from the zero point. Thus, we can now obtain the next prediction duration that starts where the prediction duration of step 1 ended, i.e. t = T + P. We then repeat this process in step 3 and shift the training duration by 2P. This shifting process is repeated for n number of steps until we obtain enough prediction durations to combine into a complete prediction duration the same as the one in the traditional method, i.e. N. In this method, the total data size(N) is equal to the sum of the product of the number of steps(n) and the prediction duration(P) with the training duration(T), i.e., $N = T + n \times P$.

The primary benefit of this method is that as predictions are never made too far away from the training duration, we minimise the increasing prediction error over time. However, the model must be retrained at each step, and training the model is computationally time-consuming. Ideally, we want P to be as small as possible for the highest accuracy. However, due to each step taking up computational time, we require P to be larger than each step's computational time, which is determined by the size of T. The computational time of each step is variable and cannot be predicted accurately, meaning we need a suitably significant P for this method to work. In our testing, in comparing the training methods, we used an initial training duration of 4 weeks (T=4 weeks). For the shifting method, we tested with a prediction duration of 1 hour up to 1 week.

4.2.1.1.3 Expanding training duration (Update Method)

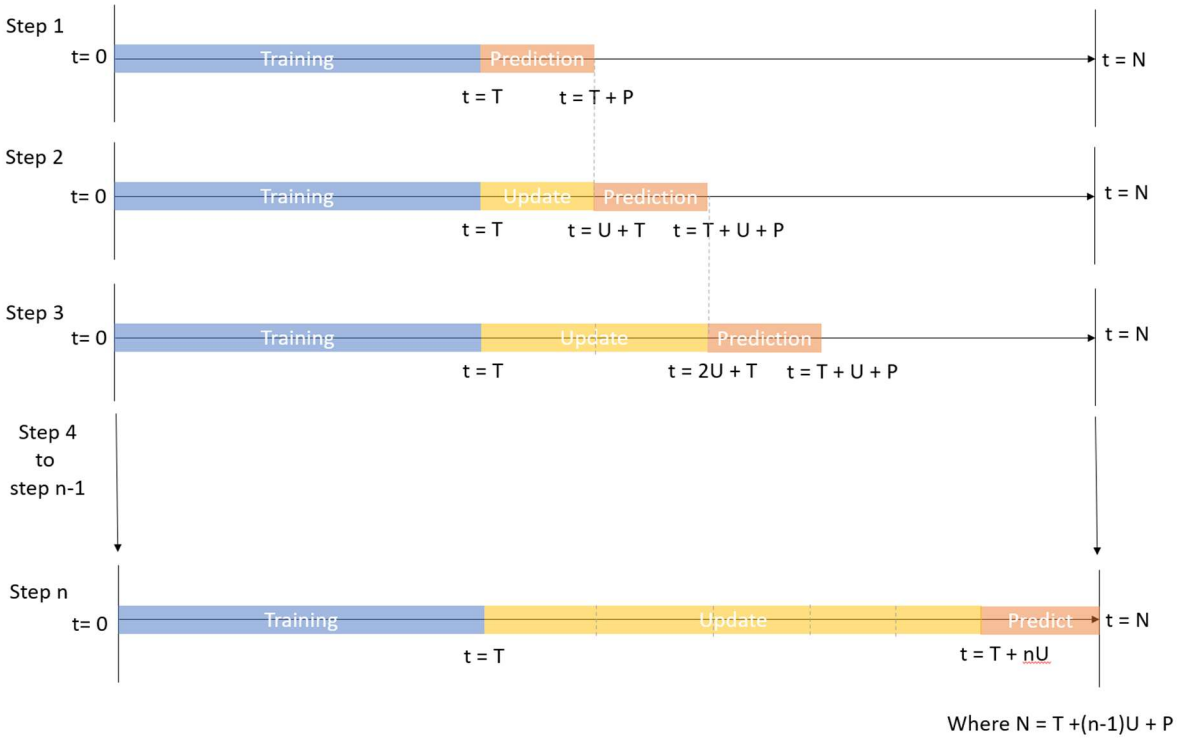

Figure 4.6 Update Method

Lastly, we look at the expanding duration method. This method introduces an update duration(U) to the model. As depicted in Figure 4.6, step 1 is identical to the shifting training duration method where the model is initially trained with training duration T, and a prediction duration of size P is made. In step 2, however, we see that instead of shifting the training duration, we add an update duration of size U between the training and prediction duration. To simplify this process, U is kept the same size as P. In step 3, we add another update duration of size U and repeat this in every step until step n.

Similarly to the shifting training duration method, we will now have multiple prediction durations starting where the previous duration ends. These prediction durations can be combined to create a complete prediction duration similar to the fixed training method. The primary difference between the expanding training duration and shifting training duration is that the data in the shifting training duration the model is trained from fresh at each step. In contrast, the expanding training duration method does an initial training once in step 1 and then performs updates to the initial training where the model continuously performs minor self-corrections. To allow for this self-correction the model needs to be continuously fed real time data which the model will back propagate into itself to perform the self-corrections within the model.

This expanding method does not require us to retrain the model at each step; instead, during the initial training, the model is built with additional inputs to allow for updates. These update steps do not require much computation, allowing us to shrink U to as low as the sampling rate. This adds the benefit of enabling us to perform the updates on the model in real-time. However, compared to the shifting method, we have a slight performance decrease in prediction accuracy. In our testing of comparing the training methods, we used an initial training duration of 4 weeks (T=4 weeks). We tested the expanding method with a prediction duration of 1 hour up to 1 week and an update duration from 1 minute to 12 hours.

4.2.2 Comparison of results from training methods

We look at testing the three different training methods described in section 4.2.1. When testing and comparing these three methods, we assess their performance using two measures. The first is the model's prediction accuracy compared with the observed data in the testing period; we quantise using the error function root mean square error between the predicted data and observed data over the testing period. The second criterion would be the computational time of each method. In this testing, we will use a fixed training period of 12 weeks and a testing period of 4 weeks. In terms of compute specifications, all training and computing were done on the same PC using a single-GPU piecewise config, with an RTX 3090 equivalent to about 16 tera floating point operations per second (TFLOP).

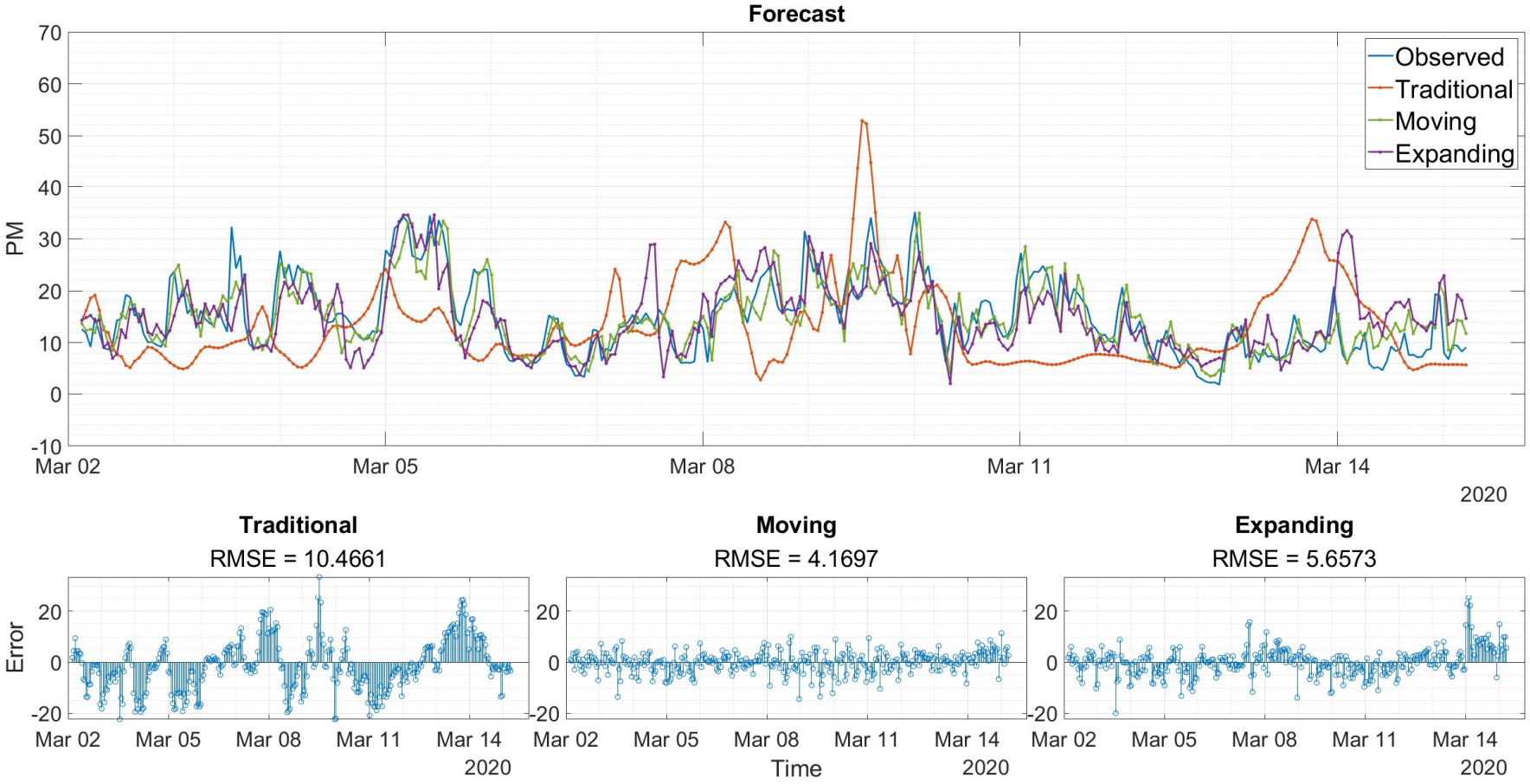

Figure 4.7 Comparison of LSTM predictions using different training methods for PM 2.5 – 2 Week Period

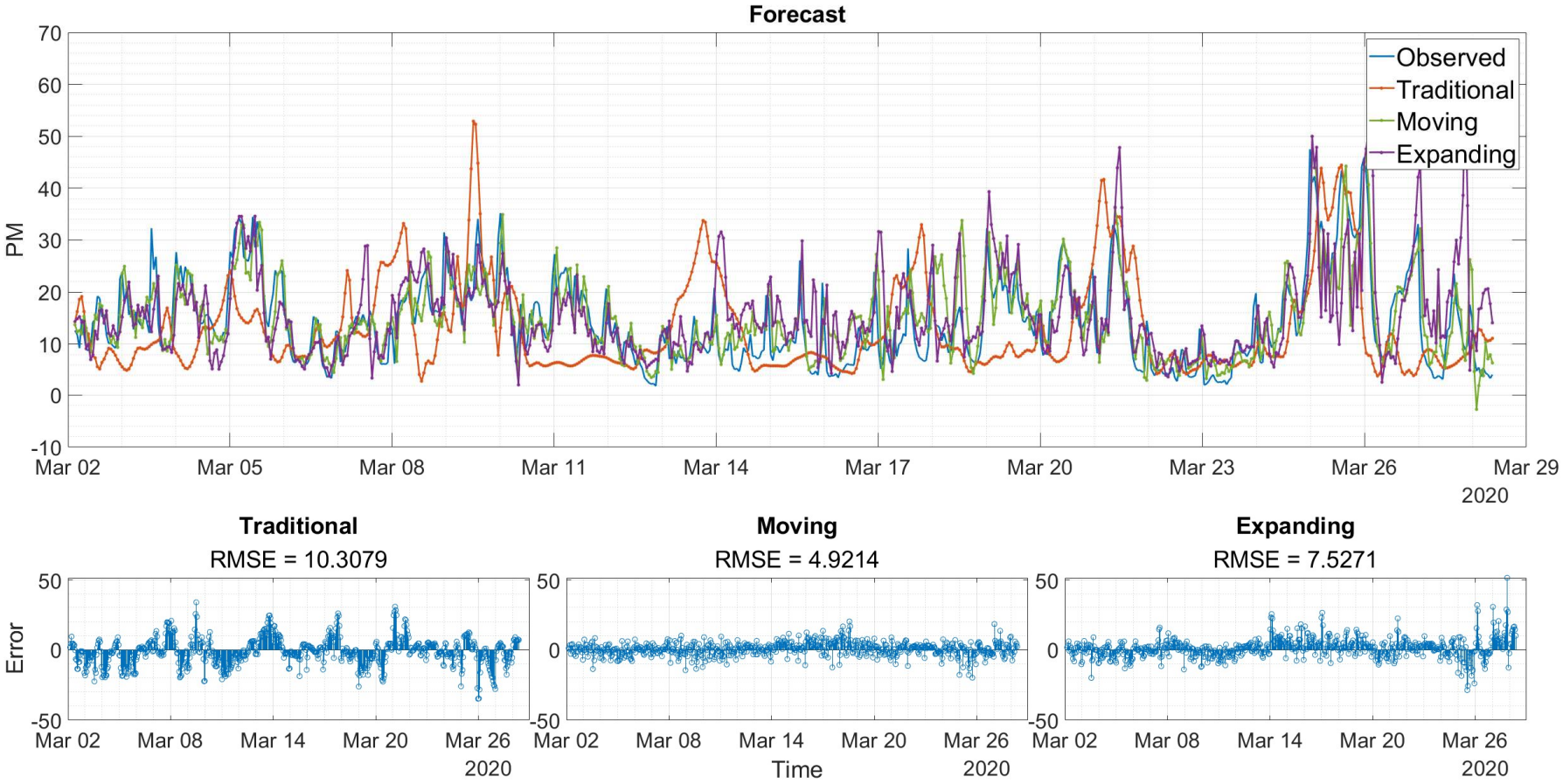

Figure 4.8 Comparison of LSTM predictions using different training methods for PM 2.5 – 1 Month Period

Figure 4.7 and Figure 4.8 shows the observed and prediction data using all three training methods for Particulate Matter 2.5. Figure 4.8 also shows the three error functions for the three training methods and their RMSE over the testing period. The computational time for each of these methods is highlighted in Table 4.2. Looking at the plots in Figure 4.8, the closest fit between the observed data and prediction is seen with the moving method followed by the update method. We can see this more clearly when comparing the error function between the predicted and observed data for each method. The moving method has the lowest root mean square error (RMSE) of 4.1697, the update method RMSE is 5.6573, and the traditional method RMSE is 10.4661. It is suspected this variation in prediction accuracy is because both the update method and moving method are provided with the observed data during the test period as well as the original fixed observed data during the training period, while the traditional method is only provided with the specified fixed amount of observed data during the training period. The discrepancy between the accuracy of the moving method and the update method is likely due to the moving method reiterating its training multiple times with the observed data in the test period. In contrast, in the update method, the newly observed data in the test period is only fed once to "update" the model.

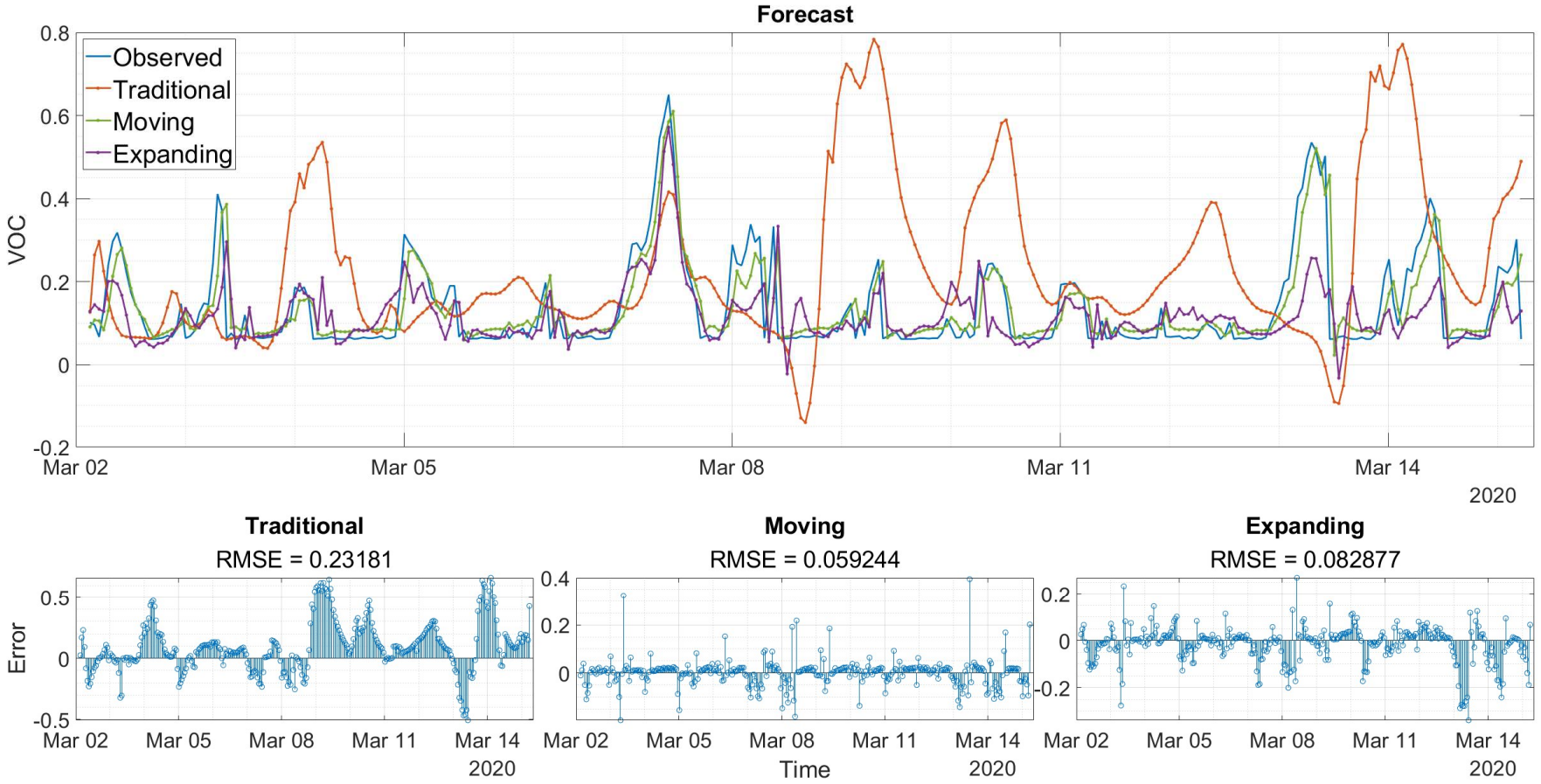

Figure 4.9 Comparison of LSTM predictions using different training methods for VOC – 2 Week Period

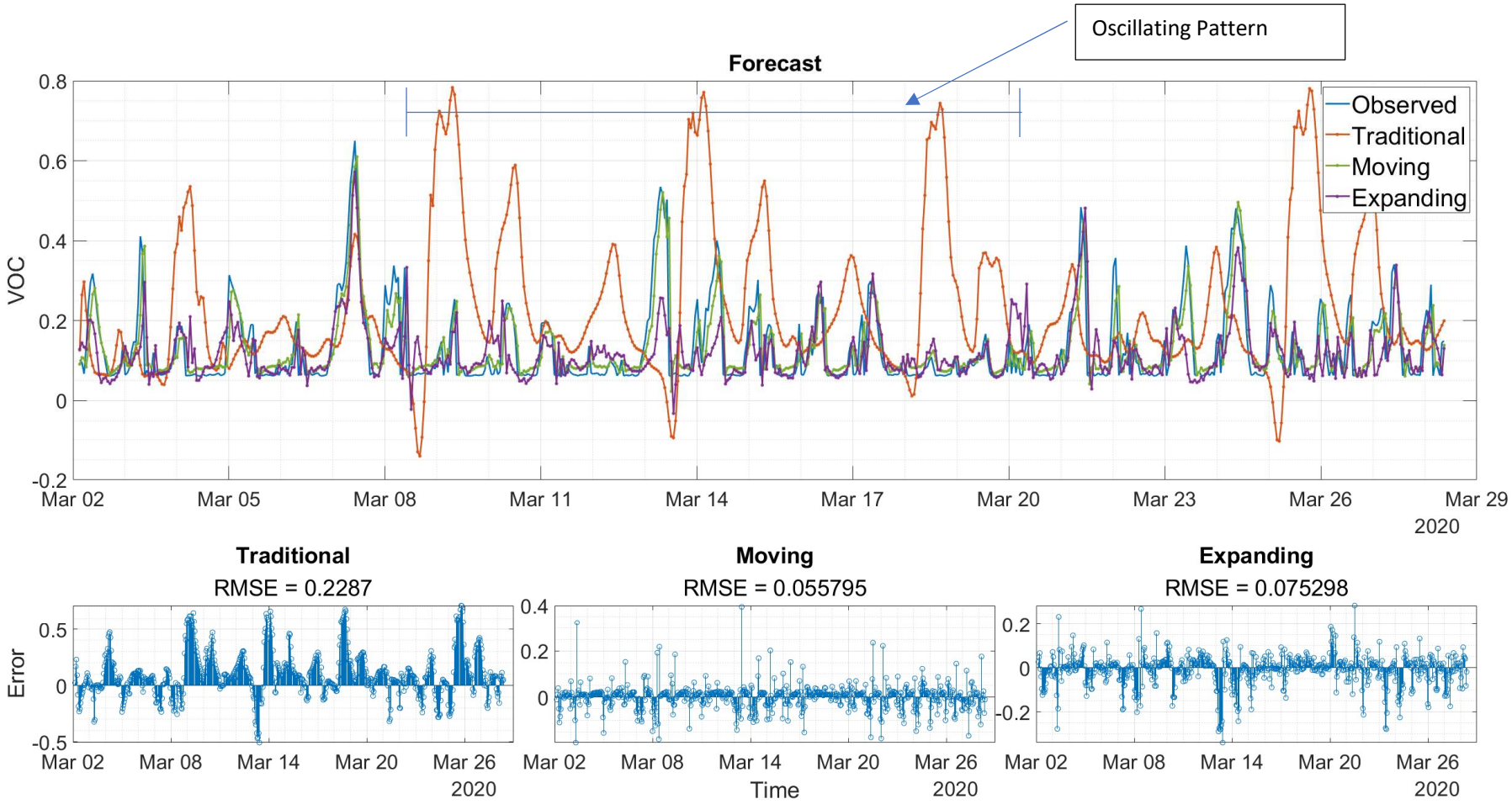

Figure 4.10 Comparison of LSTM predictions using different training methods for VOC – 1 Month Period

Figure 4.9 and Figure 4.10 shows the observed and prediction data using all three training methods for VOC. Figure 4.9 also shows the three error functions for the three training methods and their RMSE over the testing period. Looking at the plots in Figure 4.9, the closest fit between the observed data and prediction is seen with the moving method followed by the update method. We can see this more clearly when comparing the error function between the predicted and observed data for each method. The moving method has the lowest root mean square error (RMSE) of 0.059244, the update method RMSE is 0.082877, and the traditional method RMSE is 0.23181.

We also observe with the traditional training method; the predictions form an oscillating pattern from roughly March 8th. The traditional training method also only shows a rough fit to the observed data from March 2nd till March 4th; past this point, no observable fit between the predictions of the traditional method with the observed data and a loose oscillating pattern forms in the predictions. This oscillating pattern is marked in Figure 4.10. We suspect that these Oscillations are due to the model thinking it has identified a pattern in the VOC from the initial training period. However, with the traditional method the model is not aware of any changed in the environment, due to not being given any new data since the end of the training period (March 1st in this case)

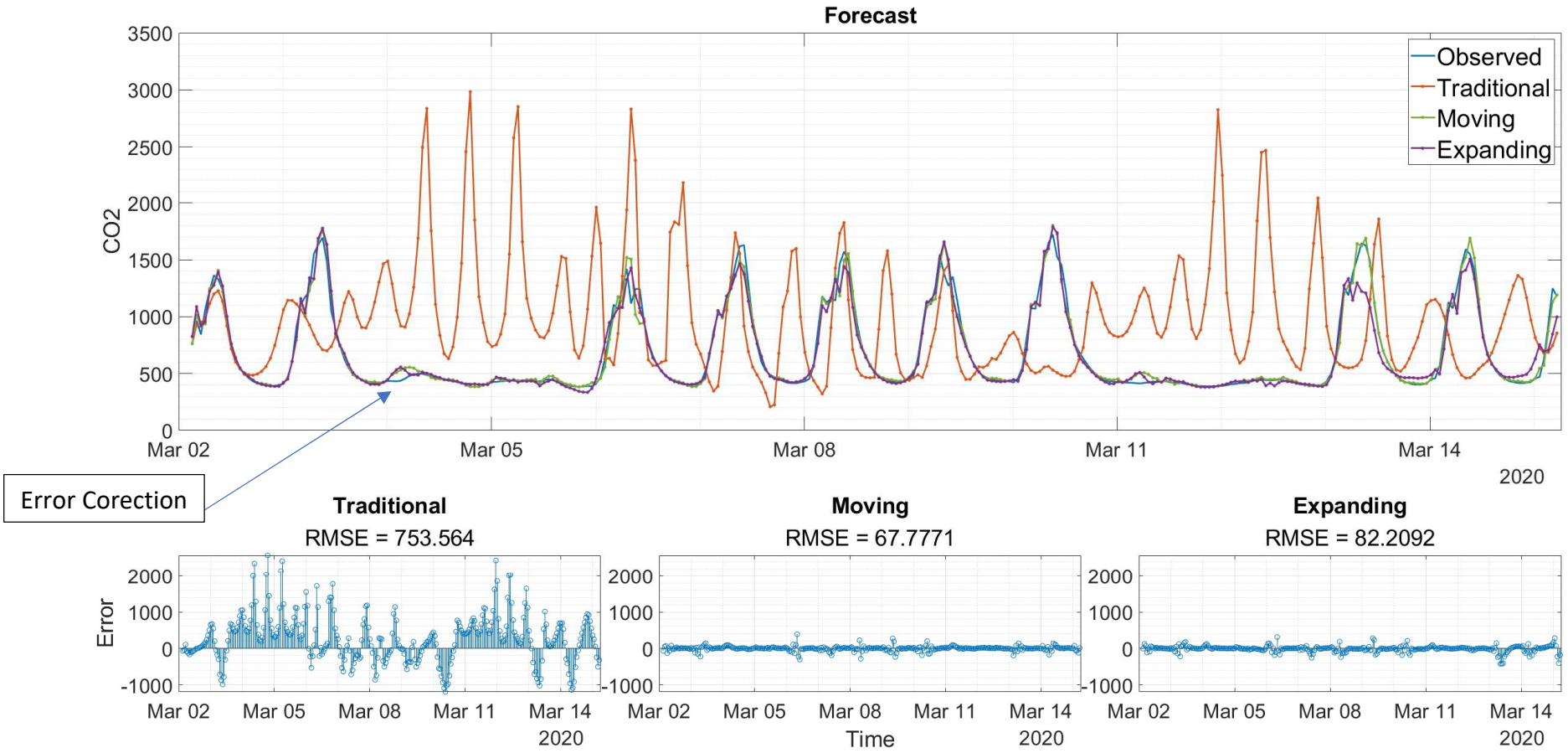

Figure 4.11 Comparison of LSTM predictions using different training methods for CO – 2 Week Period

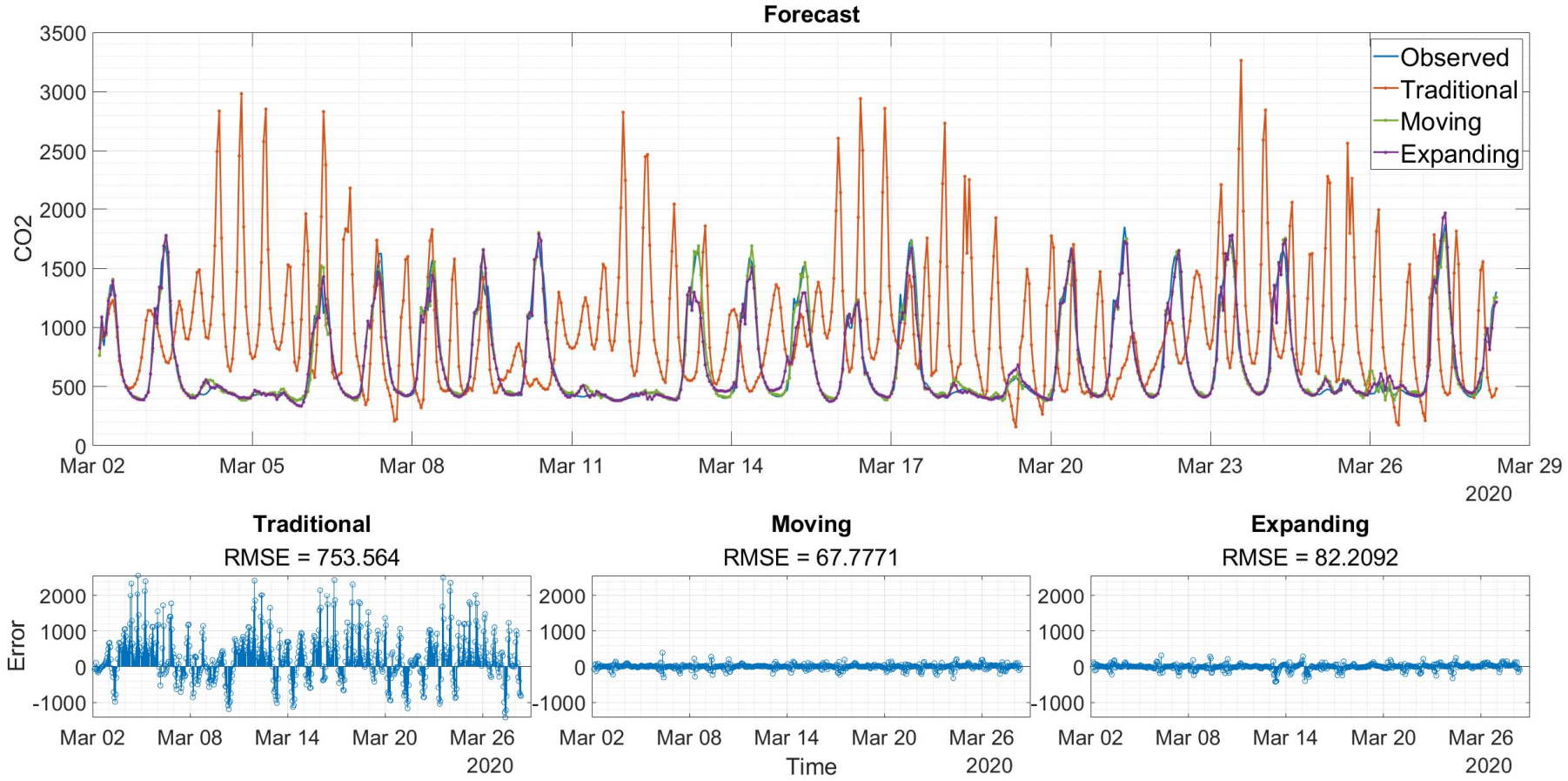

Figure 4.12 Comparison of LSTM predictions using different training methods for CO_2 – 1 Month Period

Figure 4.11 and Figure 4.12 shows the observed and prediction data using all three training methods for CO_2. Figure 4.11 also shows the three error functions for the three training methods and their RMSE over the testing period. Looking at the plots in Figure 4.11, the closest fit between the observed data and prediction is seen with the moving method followed by the update method. We can see this more clearly when comparing the error function between the predicted and observed data for each method. The moving method has the lowest root mean square error (RMSE) of 66.8291, the update method RMSE is 89.0474, and the traditional method RMSE is 721.43.

Similarly, with VOC we see a similar pattern with CO_2, where we only see a rough fit between the traditional training method and the observed data from March 2nd till March 3rd; past this point no observable fit between the predictions of the traditional method with the observed data and an oscillating pattern forms in the predictions.

This oscillating pattern, which we observed in the traditional training method of the VOC predictions and more pronouncedly in the CO_2, is likely due to the model picking up patterns in the data during its training period. This and the fact that the traditional model is unaware of any changes in the new data due to environment or other conditions. In the case of both the Moving and expanding model, this oscillating pattern doesn't form because the models get newer data, allowing them to correct any spikes they think could develop but don't in real life. In Figure 4.11 we show an area marked with error correction that is likely this exact situation happening where the moving and expanding model begin forming a spike but after a small delay(the forecast period) the models correct this spike and converge back towards the observed data

	Traditional	Moving	Expanding
PM2.5	3m 22s – 10.3079	40m 02s – 4.9214	7m 58s – 7.5271
VOC	2m 48s – 0.23181	38m 38s – 0.059244	8m 17s – 0.082877
CO_2	3m 16s – 731.43	45m 13s – 66.8291	9m 15s – 89.0474

Table 4.2 Comparison of computational time and RMSE for 2-week period for each training method

Table 4.2 shows a summary of the training times for each model for each pollutant in a week prediction period with their respective RMSE. We see the same pattern with all the 3 pollutants, where the traditional method consistently has the lowest training time but the highest RMSE, while the Moving method consistently has the lowest RMSE but a much higher training time. The expanding method sits in between but with an RMSE allot closer to the moving method while keeping a training time significantly closer to the traditional method. As such, in terms of our application of air quality predictions, we believe the expanding method to be the most suitable for its balance of accuracy and computational speed.

4.2.2.1 Moving Method - Testing Different Predictions windows (Shifts)

Here we look at testing the affects varying different prediction windows in the moving method and how it affects both the compute time and accuracy of the model. As highlighted in Section 4.2.1.1.2 the size of the prediction window used is equivalent to the shift of the training duration per step. We tested the prediction windows from using the smallest possible duration of 1 minute as this was the sample rate of the data, up to a duration of 24 hours (1440 minutes).

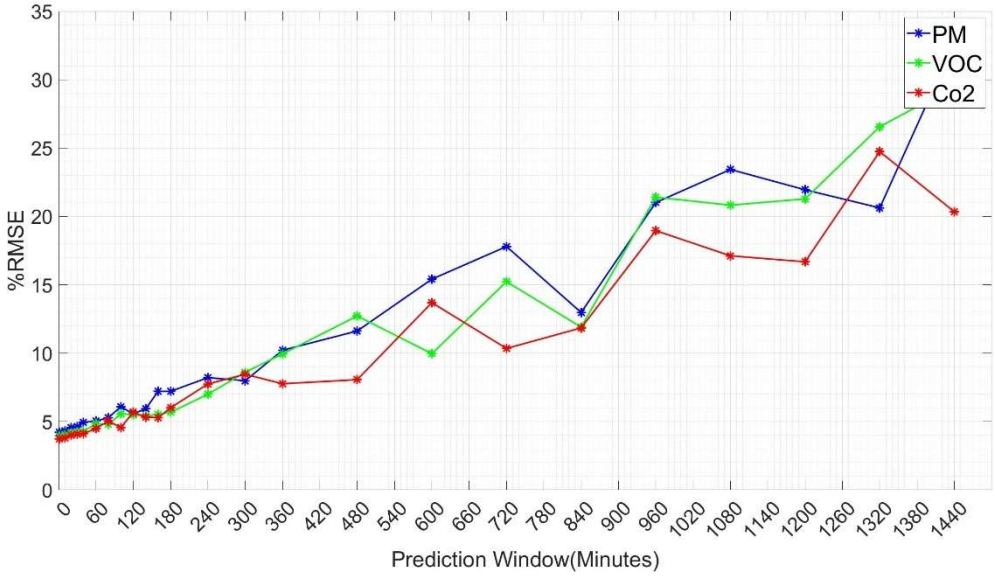

Figure 4.13 Moving Method - %RMSE when Varying Prediction Window

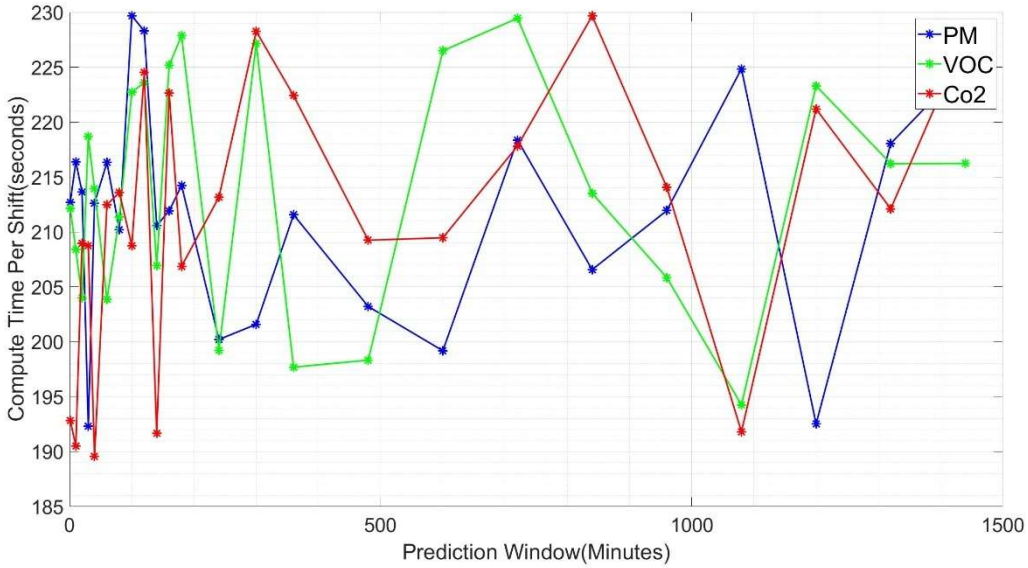

Figure 4.14 Moving Method - Compute time per shift at different prediction window.

Figure 4.13 shows the %RMSE at different prediction windows for each Pollutant. We see that the %RMSE slowly rises as we increase the size of the prediction window, where is hits the 10% RMSE mark at the 360-480 minute mark depending on the pollutant. It was also observed in Figure 4.14 that the compute time per shift is also within a fixed range of 190 – 230 seconds(roughly 2 – 4 minutes) regardless of the size of the update duration. The magnitude of this computational time graph will vary based on computational power of the system used thus if there are hardware changes on final application this analysis will need to be repeated. In order to achieve real time predictions, we require the predictions to be made/calculated before the next prediction window begins. As such the smallest possible prediction window feasible would be limited by the compute time, in the current setup this would result in a smallest possible prediction windows of 2 – 4 minutes, before accounting for any additional tolerance.

4.2.2.2 Expanding Method - Testing Different Update Durations

Here we look at testing the affects varying different update durations for the expanding method. We tested the expanding method using the smallest possible update duration of 1 minute as this was the sample rate of the data, up to an update duration of 24 hours (1440 minutes).

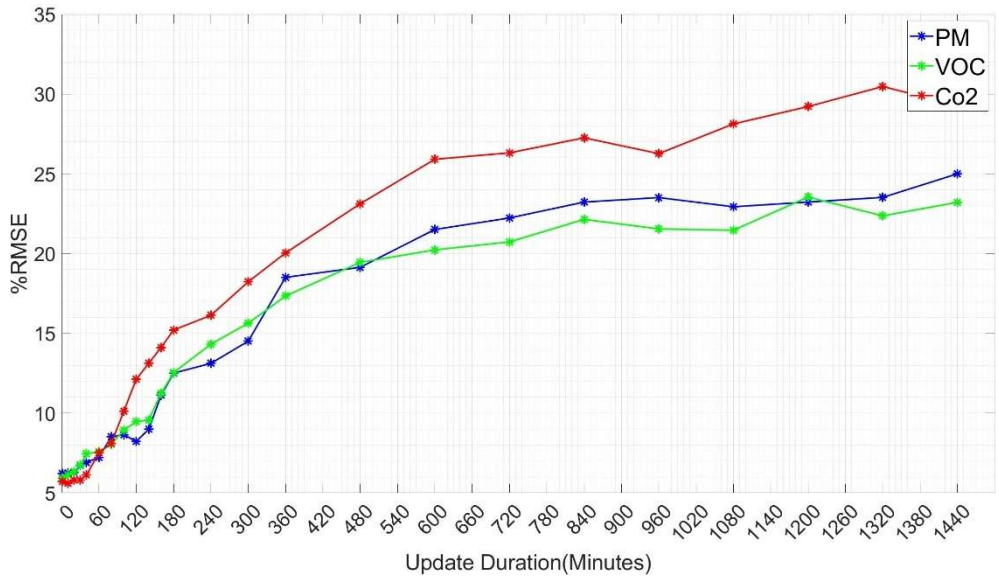

Figure 4.15 Expanding Method - %RMSE when Varying Update Durations

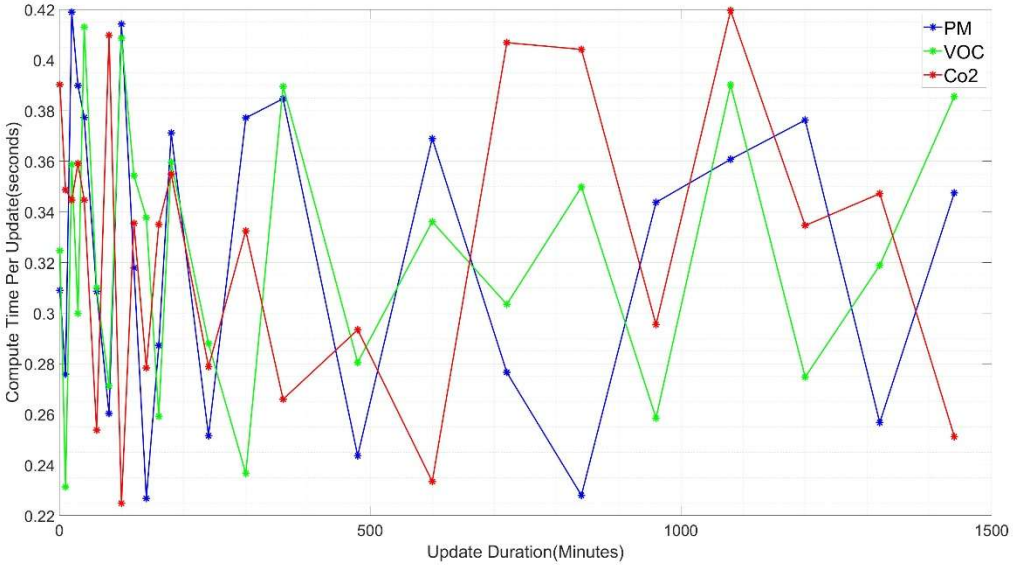

Figure 4.16 Expanding Method - Compute time per update at different update durations.

Figure 4.15 shows the %RMSE at different Update durations for each Pollutant. We see that the %RMSE breaks the 10% error mark at about 60–180 minute update durations depending on which pollutant is looked at. However, it was also observed in Figure 4.16 that the compute time per update duration is the same regardless off the size of the update duration. This will graph will vary based on computational power thus if there are hardware changes on final application this analysis will need to be repeated. However, the compute time per data point remains well below 1s even at update durations of 1 minute using current hardware which leaves a large amount of tolerance to reduce computational power. As such we believe that the minimum possible update duration of 1 minute is ideal.

When comparing how varying the prediction window in the moving method and the update duration in the expanding method we see that increase either of these variables in their respective methods will have a negative effect on the %RMSE with the expanding method generally having a worse %RMSE compared to the moving method but the magnitude of the each compute step in the expanding method is much lower than the moving method, resulting in allowing use to use much smaller updates in the expanding method. Currently the minimum compute time for the moving method was shown to be 2-4 minute, however later in this thesis we begin incorporating multivariate and multiple sites to the model which significantly increases the compute time of the model. The study choses to move forward with the expanding method as the accuracy performance to compute time is much more suitable for real time predictions.

4.3 Chapter Summary

This chapter began by examining the system's overall hardware architecture and the specifications of the various hardware components selected, including their models and accuracies. This system architecture provides a backbone for data collection and thus provides a large amount of data to test the outlined training method the rest of the chapter goes through. We compared how these three methods work and the variations they will provide in prediction accuracy and computational time. Method 3 of this chapter shows our approach to training the model, which aims to use a small sacrifice in prediction accuracy for a significant boost to computational time.

5 Overall Model Optimisation & Multisite model proposals.

This chapter begins by looking at methods to optimise the model using various known methods. We will be looking at different ways of improving prediction accuracy and the characteristics of LSTM that need to be tweaked to make it suitable for air quality predictions. This chapter mainly provides a foundation to build on for the next Chapter, which is the main novelty of my work. Following basic optimisation, we dive into looking at linking models of multiple sites to optimise further and improve the performance of the model. We show my novel method of incorporating data from multiple sites into a predictive model. This has historically proved challenging because air quality data is very localised. We aim to use this novel approach to have a macroscopic look at indoor air quality across multiple locations and, from this data improve the performance of air quality predictions.

5.1 Initial Optimisation

5.1.1 Model Training Optimisation

When looking at training the previously mentioned LSTM models in section 4.2 using the collected indoor air quality data, there are a few factors to consider in optimising the training process regarding accuracy and speed. The factors we look at here are as follows.

1. Training Duration (T)
    a. This refers to the duration of data used in the model training before any predictions are made.
    b. This applies to all 3 training methods: fixed, shifting and expanding training duration.
    c. Tests were done with training duration from 1 day up to 3 months.
2. Prediction Duration (P)
    a. This refers to the length of the prediction duration made by the model.
    b. While this duration exists in all 3 training methods, it cannot be optimised in the fixed method, and the duration will be predetermined by the size of the data set (N) and the training duration (T). As such, the optimisation of this variable was only looked at using the Shifting and Update training duration methods.
    c. Tests were done with a prediction duration of 1 minute up to 1 day.
3. Training generations
    a. This refers to the number of times the model processes the training data before it attempts to make predictions.
    b. This applies to all 3 training methods, fixed, shifting and expanding.
    c. Tests were done with training generations as low as one up to 30,000.

In each of these factors, we will look at varying the factor itself and how that affects both the model's accuracy and the computational speed. All computations are performed on the same machine to keep tests consistent. We have used single variate data in all these cases to speed up training at all iterations while testing the instances. Apart from the variables being optimised, the other 2 variables were kept constant while performing the optimisation.

5.1.1.1.1 Target %RMSE

%RMSE is the main performance benchmark selected to look at indoor air quality predictions. In selecting a cutoff point for an acceptable %RMSE we need to consider a few things. Mainly %RMSE is just a numerical indication of how accurate the model is compared to the real data, the main determining factor of how the model performs is if the model data fails to show spike in air quality data.

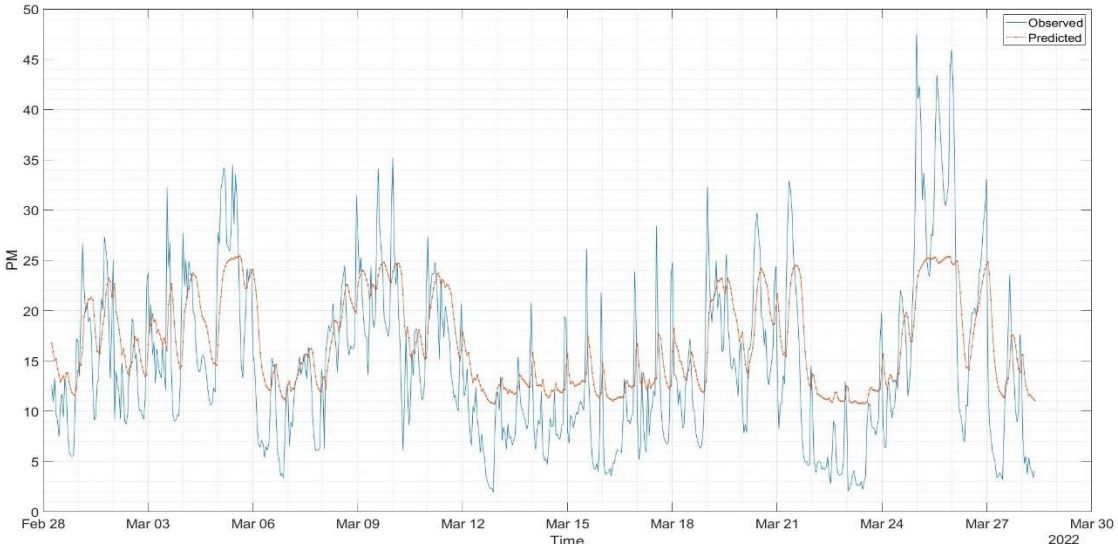

Figure 5.1 17% RMSE - Peaks and troughs less visible

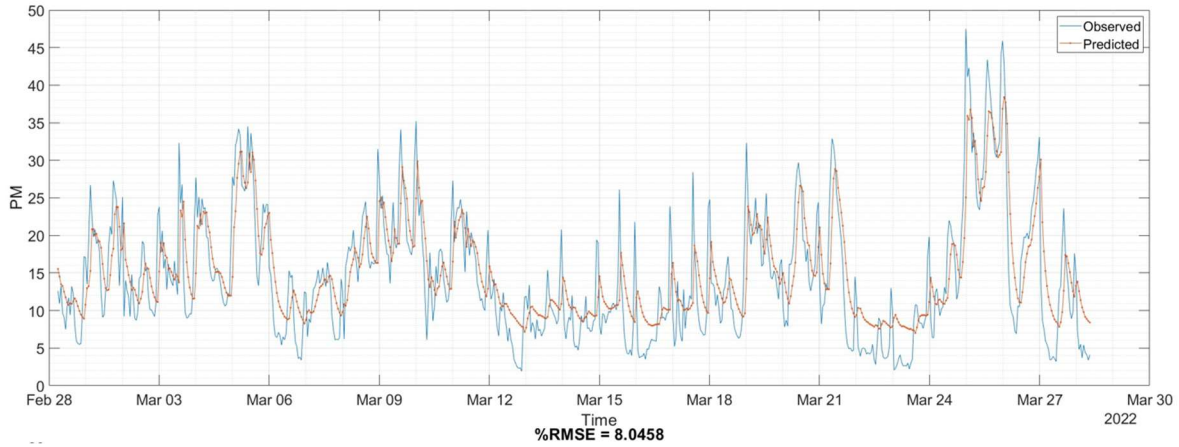

Figure 5.2 8% RMSE - Peaks and troughs still somewhat visible

We see in Figure 5.1 and Figure 5.2 the difference in the visibility of peaks and troughs when we look at a prediction with 8% RMSE and 17% RMSE. In Figure 5.2 we still can clearly see when there are spikes in air quality while in Figure 5.1 the spikes appear much more muted. As such we initially choose to use a 10% %RMSE as the largest acceptable error. To further reinforce this other studies which were highlighted in sections 3.2.7, 3.3.1 and 3.3.2 also use 10% %RMSE as their cut off point. Furthermore, this study will later further improve the %RMSE using other methods, and some evaluations are done with reduced parameters to reduce computational time.

5.1.1.2 Training Duration

We started the optimisation by looking at the training duration. When testing various training durations look at testing the training duration we kept the other training factors constant. In this case, we had kept the prediction duration to 2 hours and the training generations to 300. The following tests are all performed using the expanding method, as we believe this method is the best for our application.

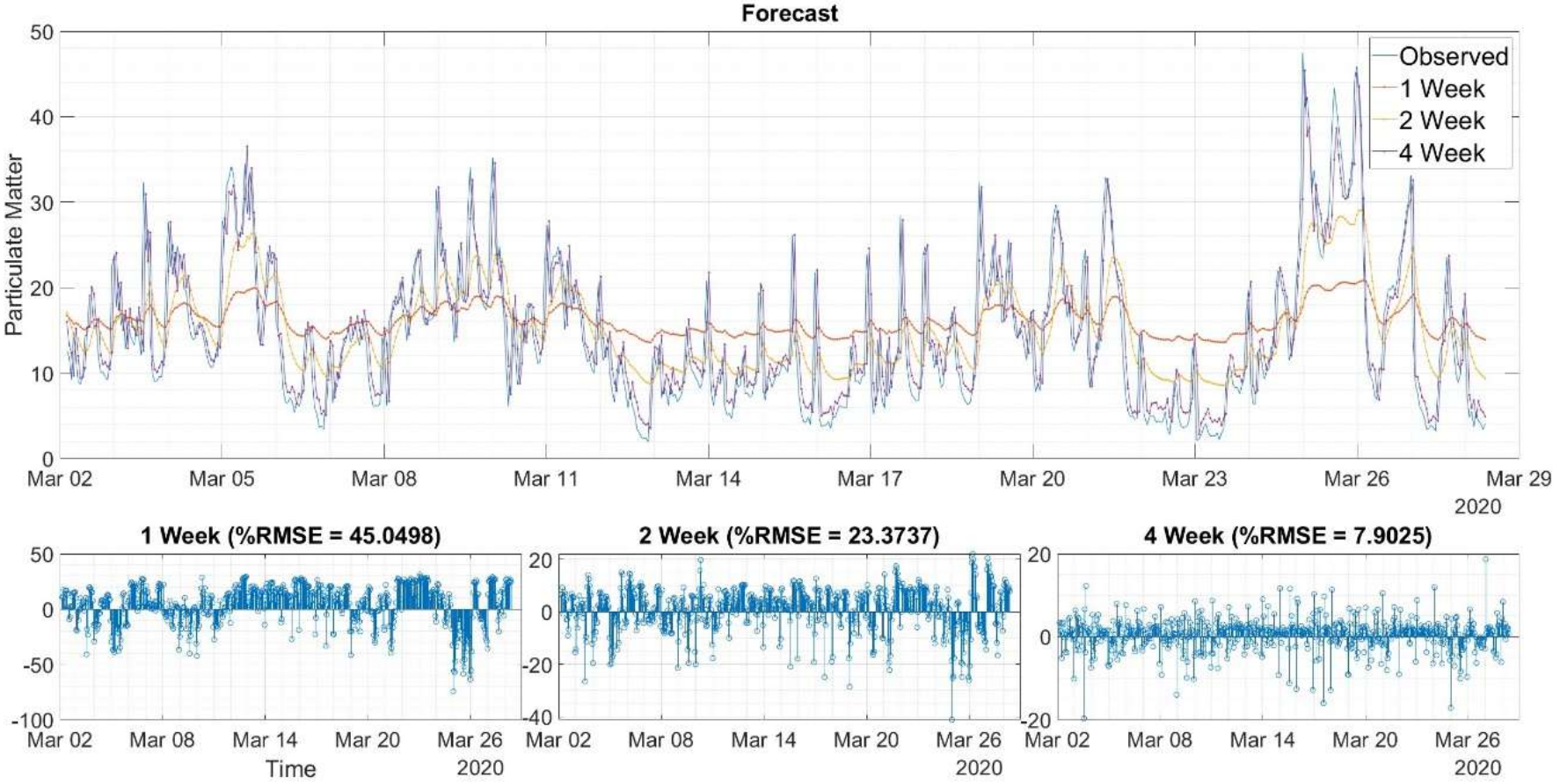

Figure 5.3 PM 2.5 Comparing Training Durations

Figure 5.3 shows an initial comparison of the PM 2.5 level predictions. In this initial comparison, we used training durations of 1 week 2 weeks and 4 weeks, shown with the orange, yellow and purple lines, respectively. At the bottom of the figure, we have shown the error function of each of the predictions compared to the observed data and their %RMSE, which we use as a numerical measure for the performance of the predictions. We see in Figure 5.3, the %RMSE increases from 45.0498 to 23.3737 and finally 7.9025 with 1 week, 2 week and 4 week training respectively. This implies a larger training duration provides a more accurate prediction.

Further, using the %RMSE of the predictions, we can plot a graph of this %RMSE at various training durations to further assess how the training duration affects the prediction accuracy.

To further test the effects of the training duration on the prediction accuracy, we will proceed by plotting the RMSE over a 2-week testing period using varying training periods from 0.5 weeks to 12 weeks. This test will be repeated with data from 7 sensors which we distributed amongst 3 sites at varying locations.

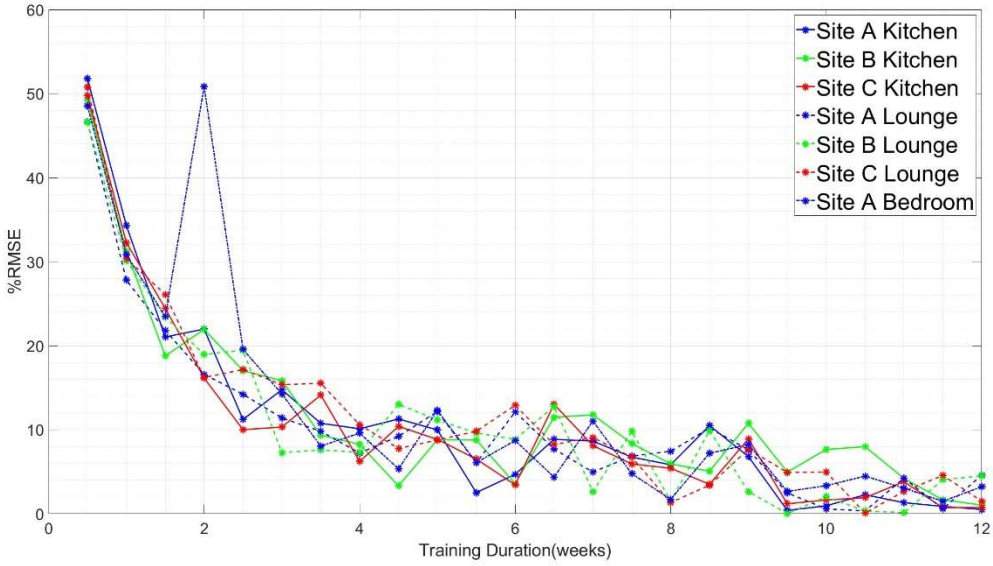

Figure 5.4 PM 2.5 Training Duration Optimisation

Figure 5.4 shows how the %RMSE varies as we change the training duration of the model for particulate matter. We can see from this that the performance increase is most significant up to week 2, while we still see significant improvements up to week 3. After that point, we gradually get diminishing performance improvements as we increase the training duration.

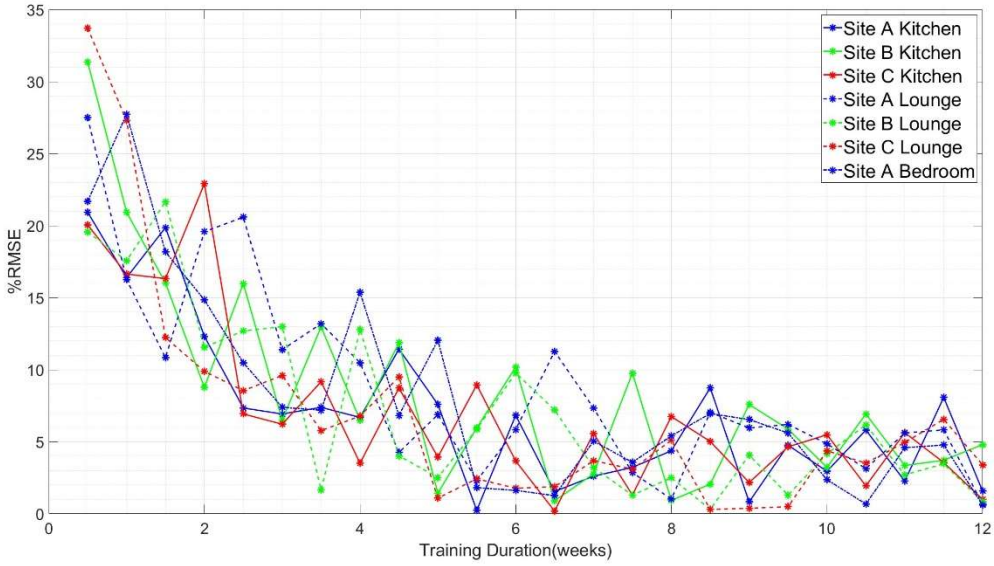

Figure 5.5 VOC Training Duration Optimisation

We repeated the same test looking at VOC instead, shown in Figure 5.5. Here we see a similar pattern where the performance increase diminishes as we use larger and larger training durations. However, in the case of VOC, we notice that the initial %RMSE on the VOC prediction at even one week is lower than the %RMSE in the case of PM. We also can observe that the %RMSE approaches the 10% mark at about four weeks but stays consistently below the 10% mark after week 7 and 8.

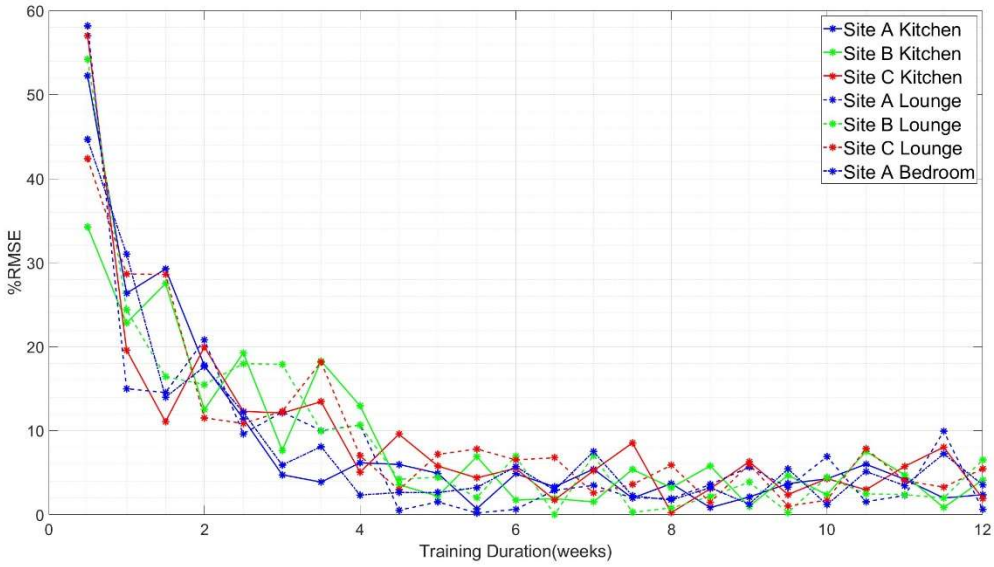

Figure 5.6 Carbon Dioxide Training Duration Optimisation

Figure 5.6 shows the same optimisation test but looks at Carbon dioxide instead. Here we observe the repeating pattern of the first four weeks leading to the most improvement, but in the case of Carbon Dioxide, little to no progress can be seen past the 5/6-week mark.

5.1.1.3 Prediction Duration

We performed a similar test as with the training duration with the prediction duration, but instead of varying the training duration, we changed the prediction duration while keeping all other variables constant. For this test, we kept the training generation at 300 and the training duration at six weeks.

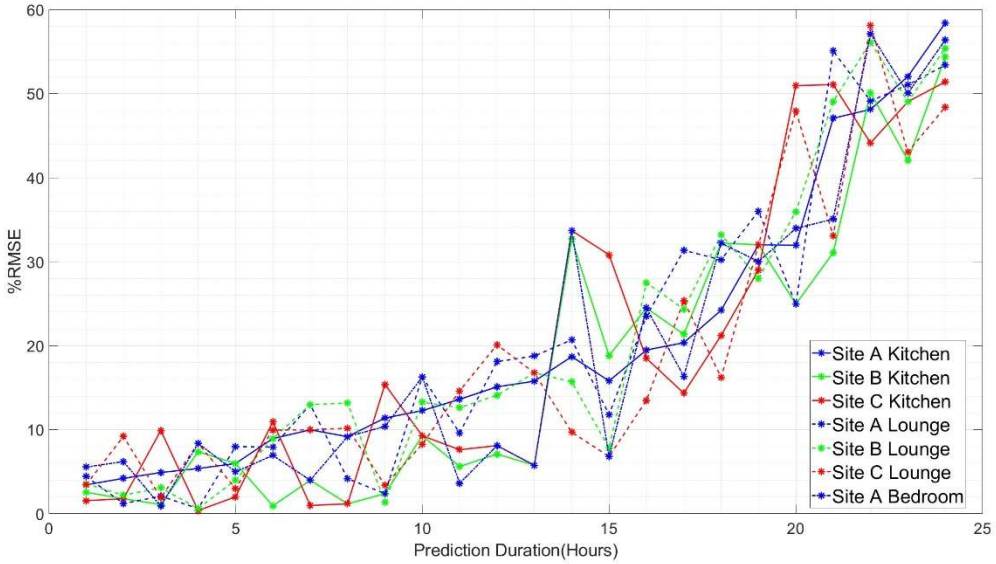

Figure 5.7 PM 2.5 Prediction Duration Optimisation

Figure 5.7 shows the effect of an increasing prediction duration on the %RMSE of the model for PM 2.5, which indicates the model's accuracy. We can see that the model's accuracy gets worse as we increase the prediction duration. Values past the 5-hour mark start to surpass the 10% mark. Another observation is that past 20 hours, the %RMSE looks like it may be plateauing. We suspect this is because the error is as high as it can be while still within the limits of the possible readings.

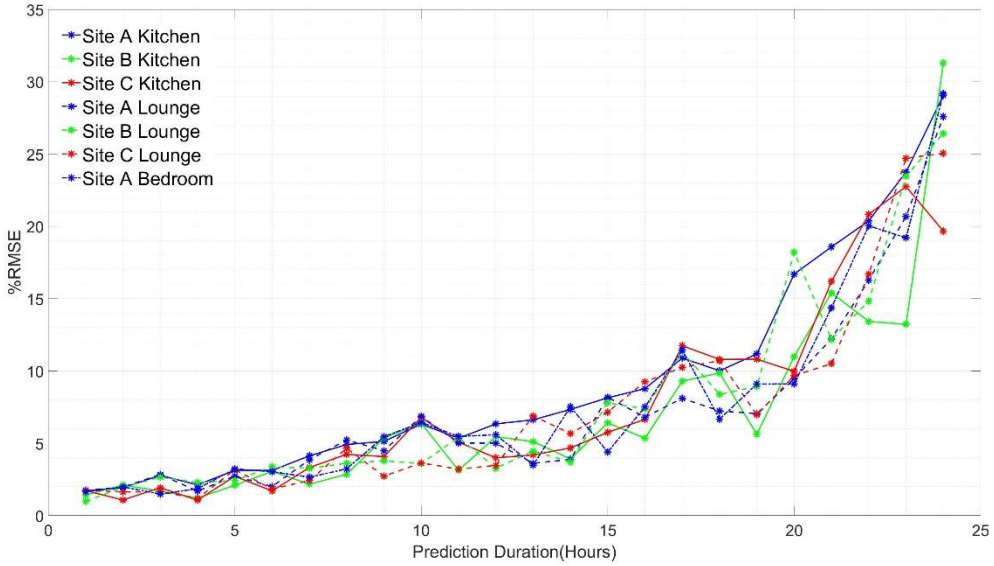

Figure 5.8 VOC Prediction Duration Optimisation

Figure 5.8 shows the effect of an increasing prediction duration on the %RMSE of the model for VOC, which in turn indicates the accuracy of the model. We can see that the model's accuracy gets worse as we increase the prediction duration. Values past the 9-hour mark start to surpass the 10% mark. Compared to PM 2.5, we see a more accurate prediction for longer. We suspect this is due to the random nature of PM 2.5 readings, which we can see when we compare Figure 4.8 and Figure 4.9. In these figures, we see that VOC has some form of recurring pattern while PM 2.5 is almost completely random

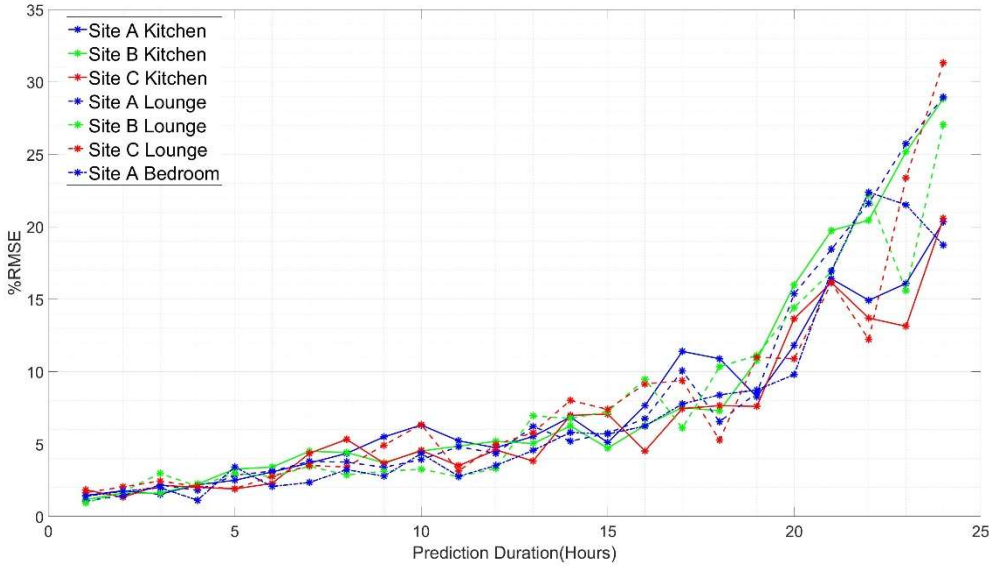

Figure 5.9 Carbon Dioxide Prediction Duration Optimisation

We can see in Figure 5.9 that Carbon Dioxide behaves very similarly to VOC. This is likely because the Carbon Dioxide sensor being used is an eCO_2 sensor which is linked to the VOC sensor.

5.1.1.4 Training Generations

We performed a similar test with the training duration and the prediction durations for the training generations. We changed the number of training generations while keeping all other variables constant. For this test, we kept the prediction duration at 3 hours and the training duration at six weeks.

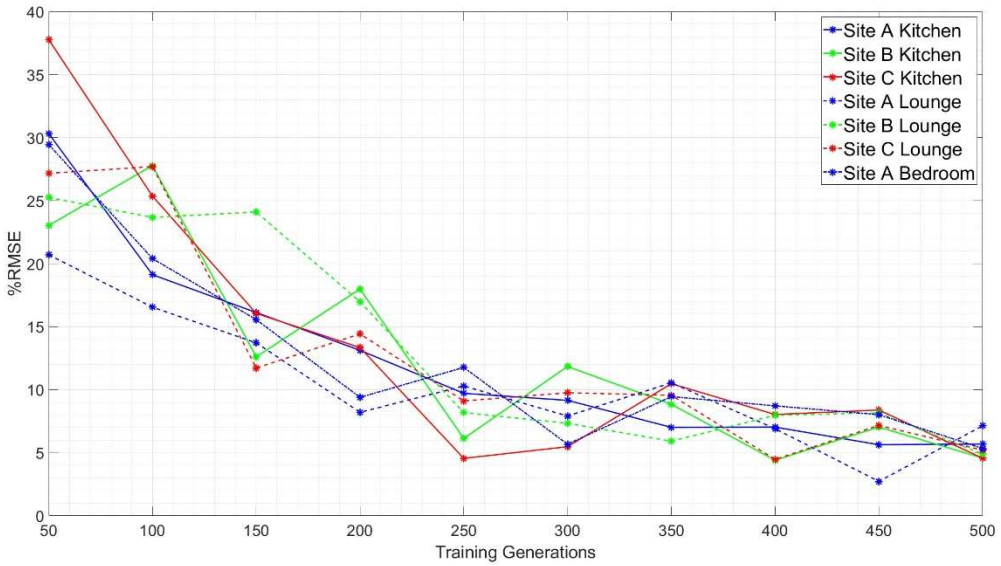

Figure 5.10 PM 2.5 Generations Optimisation

Figure 5.10 shows how the number of training generation affect the %RMSE for PM 2.5. We see that most performance improvement happens up to the 250-350 generation mark. We still get improvement in performance past this point, but the returns are less significant.

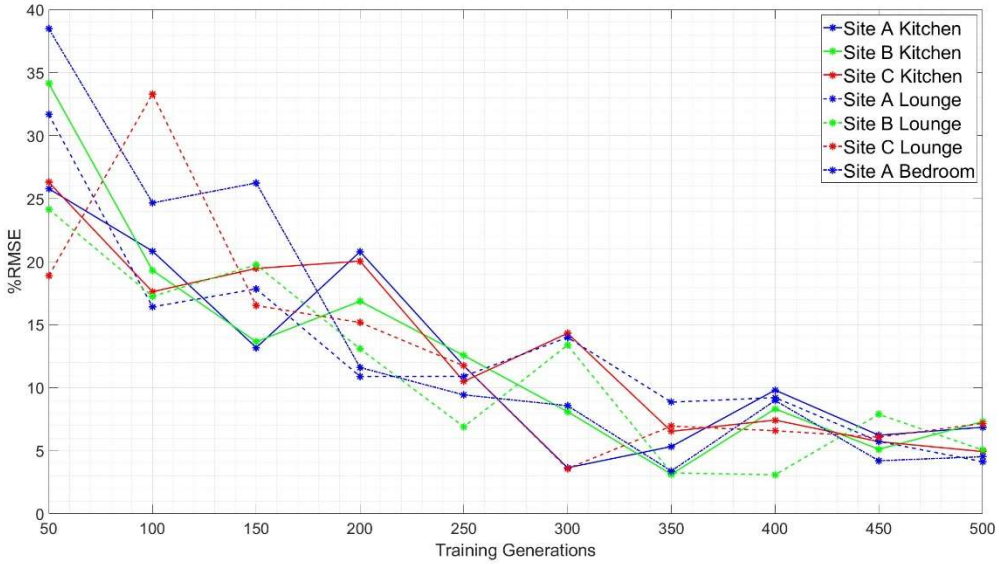

Figure 5.11 VOC Generations Optimisation

Figure 5.11 shows how the number of training generation affect the %RMSE for VOC. We see that most of the performance improvement happens up to the 250-350 generation mark. We still get improvement in performance past this point, but the returns are less significant.

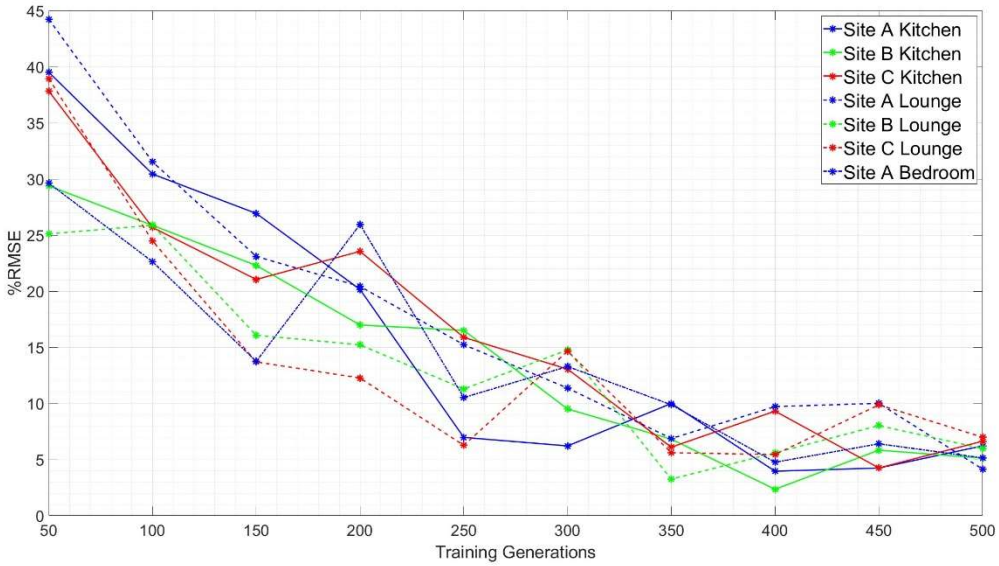

Figure 5.12 Carbon Dioxide Generations Optimisation

Figure 5.12 shows how the number of training generation affect the %RMSE for carbon dioxide. We see that the majority of the performance improvement happens up to the 250-350 generation mark. We still get improvement in performance past this point, but the returns are less significant.

In the case of training generations, we see similar characteristics amongst all three pollutants in different rooms and different sites. This likely means that the training generations are not significantly affected by the features of the data.

5.1.2 Multivariate Predictions

Initially, we only looked at making predictions using a single air quality factor as the input and output of the model. The aim of looking at multivariate predictions is to find any correlation between the air quality factors. These correlations would hopefully allow us to assist in making further predictions more accurate.

5.1.2.1.1 Pearson R score

Initially, when looking at multivariate, we looked at a linear regression model between each combination of pollutants. For each combination, we obtained a Pearson R Score, the covariance of the two variables divided by the product of their standard deviations. This is done by using Pearson R equation：

$$R = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\sqrt{\Sigma(x-\bar{x})^2(y-\bar{y})^2}}$$

5.1. We repeat this calculation for every combination of pollutants.

Pearson R equation：

$$R = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\sqrt{\Sigma(x-\bar{x})^2(y-\bar{y})^2}} \quad 5.1$$

Where?

R =correlation coefficient

x =values of the x-variable in a sample

$\bar{x}$ =mean of the values of the x-variable

y =values of the y-variable in a sample

$\bar{y}$ =mean of the values of the y-variable

5.1.2.1.2 Testing based on RMSE.

As discussed in section 3.1.1, some prior evidence of correlations between air quality factors exists. As such, we tested the effect of entering different IAQ factors into a multivariate variation of LSTM. In doing so, we aimed to see how different combinations of input variables would affect the model's percentage RMSE (%RMSE). We performed tests on every combination of input variables, including all combinations of 2,3,4 and 5 input

variables. We also repeated the same test to data from multiple sites to test if the correlation varies from site to site or is fixed across all locations.

5.1.2.2 Multivariate Forecast Correlation and Forecasting

This section looks at the possible correlation between the different pollutants in a single household. In section 3.1.1, we have seen evidence of statistical correlations between the various air pollutants.

Initially, when looking at multivariate, we looked at a linear regression model between each combination of pollutants. For each combination, we obtained a Pearson R Score, the covariance of the two variables divided by the product of their standard deviations.

	Co2	Humidity	PM2.5	Temperature	VOC
Co2	N/A	0.1069	-0.2281	0.7132	0.3983
Humidity	0.1069	N/A	-0.0118	-0.0463	-0.1000
Pm2.5	-0.2281	-0.0118	N/A	-0.2337	0.0120
Temperature	0.7132	-0.0463	-0.2337	N/A	0.2150
VOC	0.3983	-0.1000	0.0120	0.2150	N/A

Table 5.1 Linear Correlation Coefficient or each Factor combination

Table 5.1 shows the Pearson R score for each combination of the pollutants. As most of the varieties have an R score of less than 0.4, they can be considered to have a very weak linear correlation. The exception to this is temperature and CO_2, shown in the table as highlighted in green, indicating a strong correlation.

We then looked at applying the data once again into an LSTM Neural Network, but this time using multivariate data to train the model. We also fed the trained model updated data from earlier results and an 8-week training period as from our previous testing in section 5.1.1.2, the 7 to 8 week point is where the error drops below the 10% point

We then looked at applying the data once again into an LSTM Neural Network, but this time using multivariate data to train the model. We also fed the trained model updated data from earlier results and an 8-week training period as from our previous testing in section 1.2.1, the 7 to 8 week point is where the error drops below the 10% point.

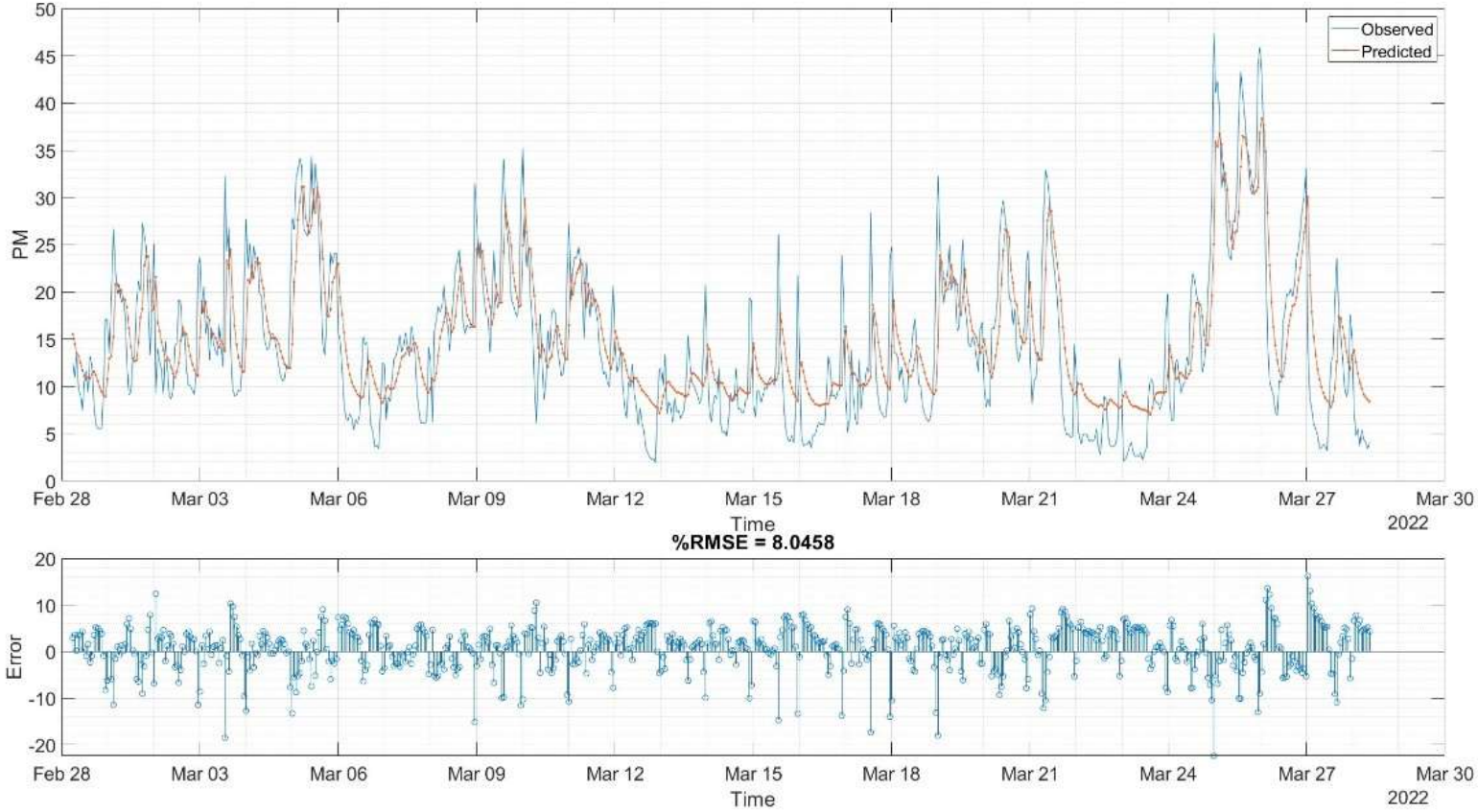

Figure 5.13 Singlevariate LSTM Predictions

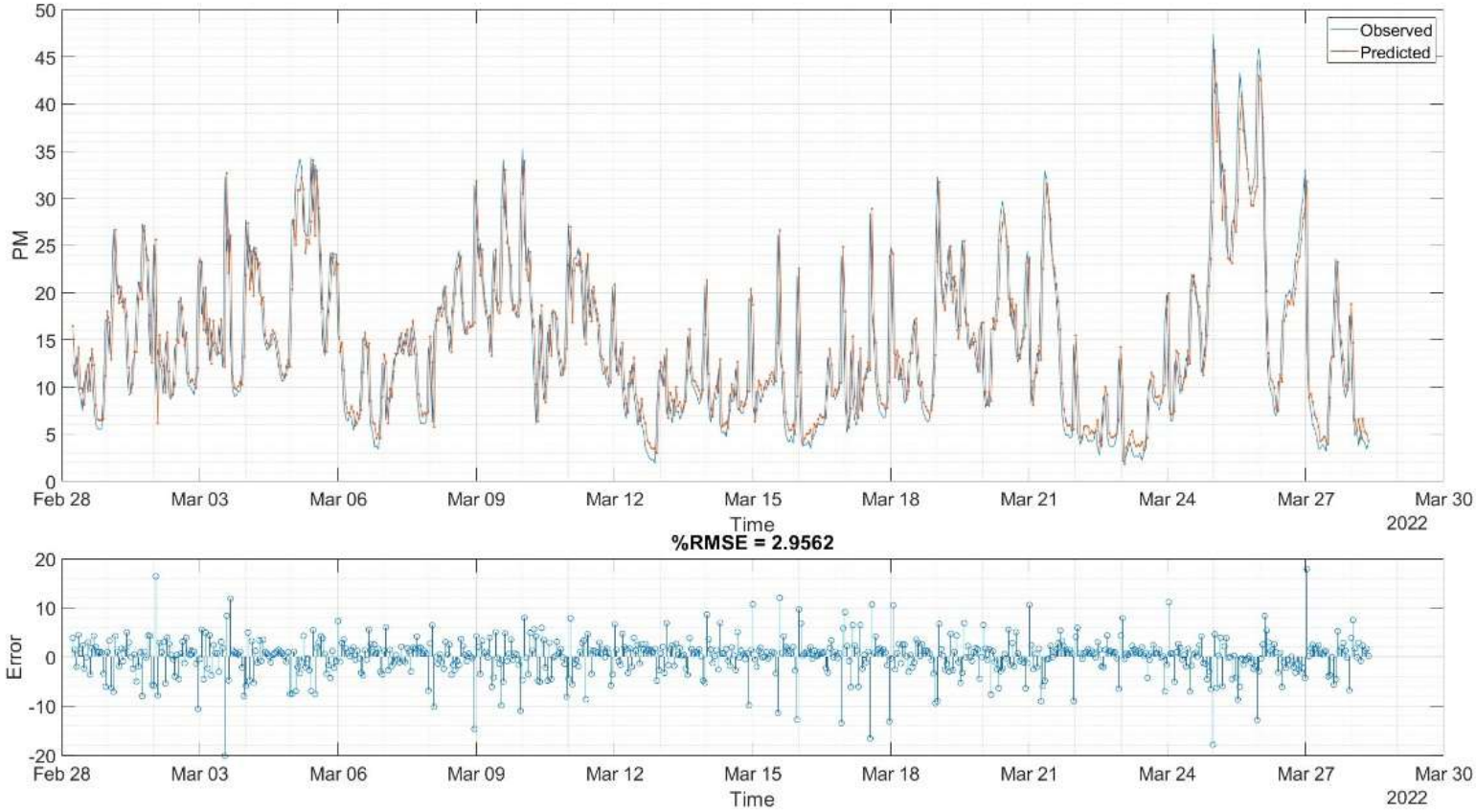

Figure 5.14 Multivariate LSTM predictions.

Figure 5.13 shows the predictions obtained from the model when it has been fed with just the PM data. Figure 5.14 shows the forecasts obtained from the model when it fed all the pollutant data as inputs. When all the pollutant data was fed as an input to train the model, we can see that the %RMSE has been reduced from 8.0458 to 2.9562. This would indicate that the different pollutants have some form of correlation with each other. This contradicts the results from the Pearson test, as the Pearson test had shown that there is only a weak or no correlation between PM 2.5 and other pollutants. As the Pearson test is a linear correlation test, this indicated that there is some form of relationship between the variables is not a linear correlation, but it is a more complicated correlation which is consistent with what was mentioned in Section 3.1.1. A few other things to note is that the multivariate LSTM model took a significantly larger computational time to train to model, at least 10x more than the single variate LSTM model. As such, the increased accuracy comes at a significantly higher computational cost. However, the computational cost is only to train the model.

Based on this, we proceeded to look at how all the different combinations of input variables affect the %RMSE of the model. We performed this with a training duration of 8 weeks, a prediction duration of 1h and 500 training generations.

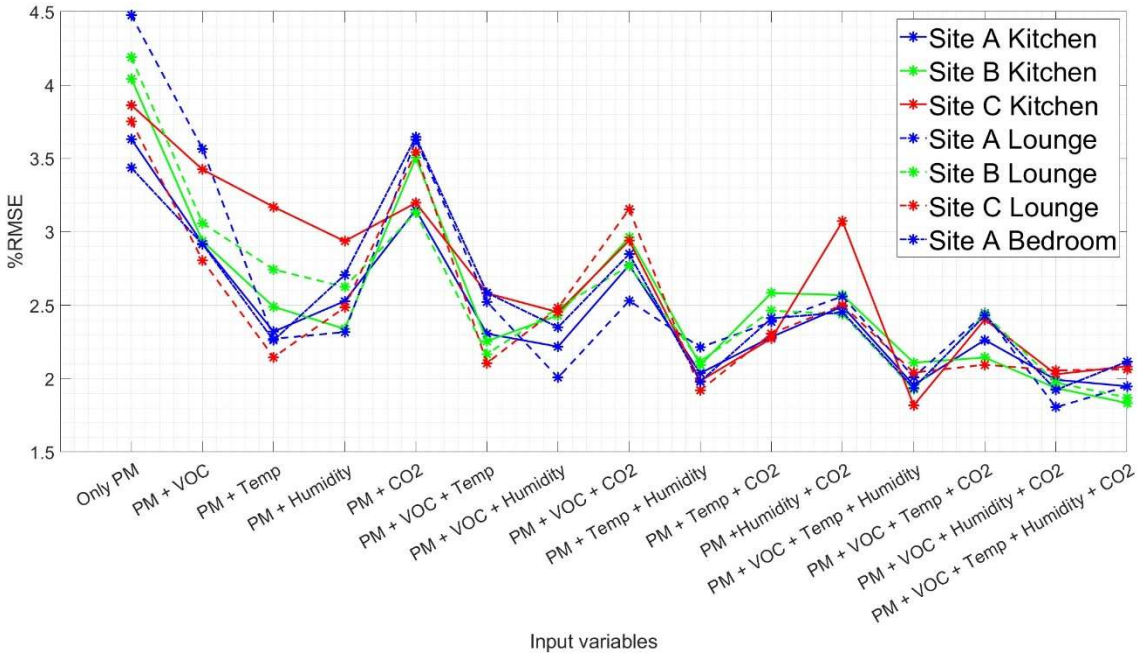

Figure 5.15 Multivariate Input variable combinations

Figure 5.15 shows the % RMSE with the different input variable combinations. As we can see, there is a general downward trend in the %RMSE as more variables are incorporated. We also observe that in general incorporating specific input variables has a larger effect on the %RMSE: temperature and humidity.

5.1.3 Hyperparameter optimisation

Within LSTM, there exists a set of hyperparameters that can affect the performance of the model based on the dataset used. Some of these hyperparameters are as follows.

- Gradient threshold
- Initial learn rate
- Learn rate drop period
- Learn rate drop factor
- Weight initialisation
- Decay rate
- Batch size

In order to identify how these parameters affected the model, we tested the performance in terms of the percentage RMSE. The characteristic such as the deformations in the shapes of the peaks and troughs, as well as a fixed upward shift of the troughs when compared to the real data. We also used different data sets from different sites to see if these parameters would need to be varied from site to site or can be fixed across all sites. For the parameters that we could fix across sites, we used these plots to find the optimal value to set these parameters too. However, some of the parameters would produce varying results during different circumstances, such as the site location or the actual time period of the data.

For said parameters that we would need to varied depending on the characteristics of the data, we worked on developing an algorithm that would test and optimise this from time to time in order to keep these parameters at the optimal values to keep the accuracy of the model as high as possible. Figure 5.16 shows a flow chart of the proposed hyperparameter optimisation algorithm. It starts by running an initial optimisation process; this optimisation process involves running the model multiple times while varying the chosen hypervariable. The optimisation process starts by lowering the number of generations and hidden states. This is done to shorten the computational time of each iteration at the cost of lower accuracy. Upon doing so, it starts testing the model using the previous optimal value for the hyperparameter that is being tested; in the case of the initial optimisation, a "generic optimal" value is used. In the next step, the algorithm would make two iterations up and down from the initial value, and the algorithm will constantly calculate a moving gradient of the RMSE against the parameter being optimised. The algorithm then continues the iterations in the direction of the negative gradient until the gradient becomes positive (across three values), taking the point with the lowest RMSE as the new optimal value. It will finally test the last few iterations again, but with the standard number of hidden states and generations.

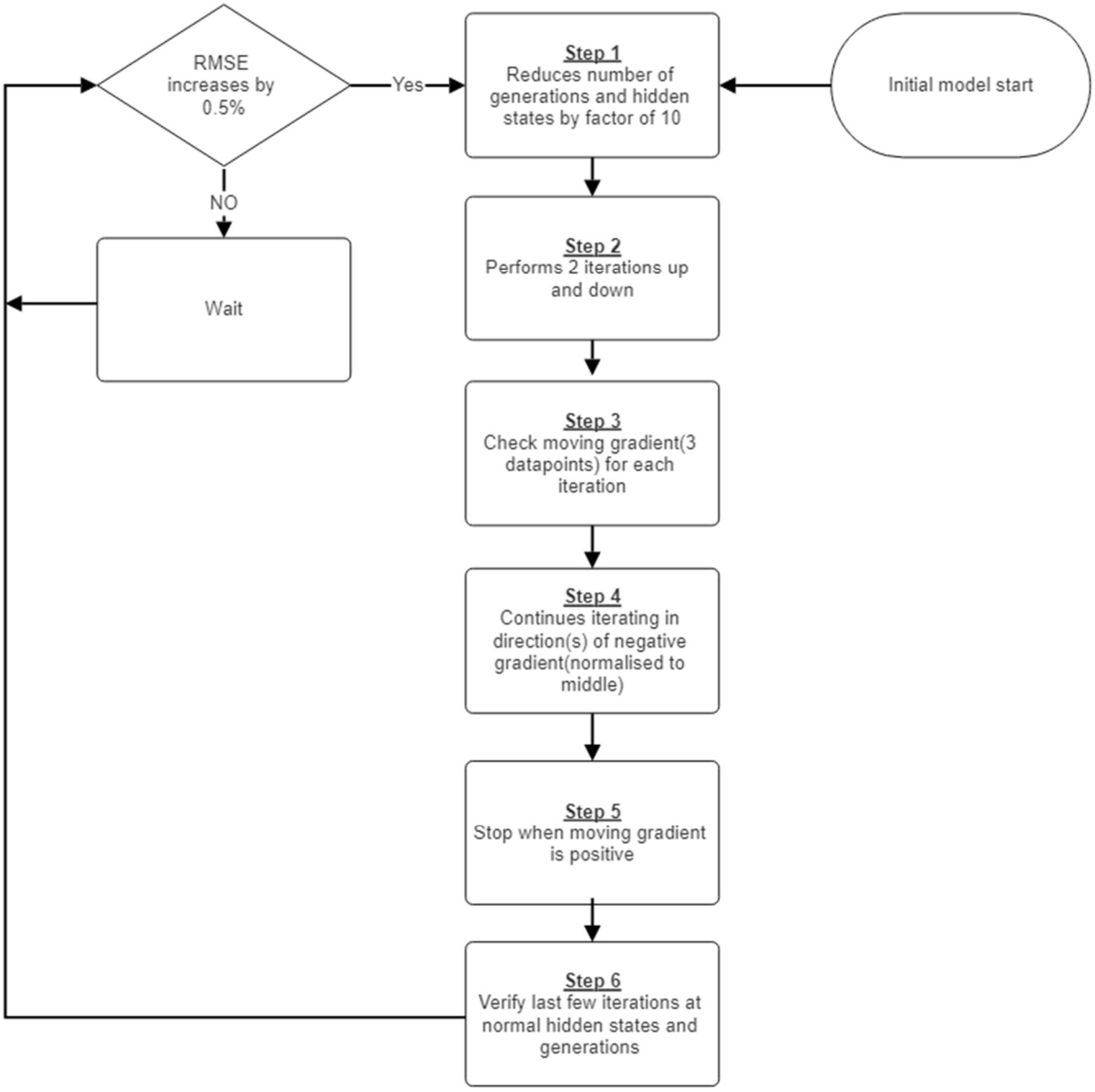

Figure 5.16 Hyperparameter Optimisation

Apart from the normal optimisation process shown in Figure 5.17, there are two unique situations we need to consider. Both situations occur in step 2, where it performs two iterations up and two down. In this step, if the gradients in both directions are positive or negative. When both are positive, the hyperparameter is already at its optimal value, and the algorithm will skip to the last step; this is shown in Figure 5.19. The second situation is shown in Figure 5.18, where both sides are negative. It will continue iterating in both directions until it gets to a positive gradient on both sides and will then choose the side with a lower RMSE.

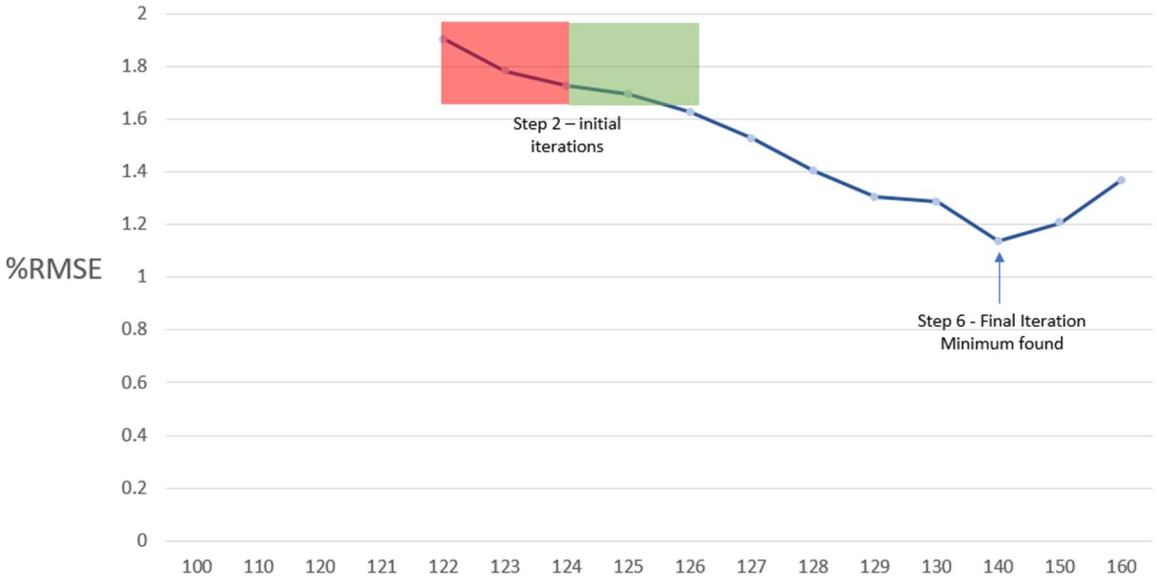

Figure 5.17 Standard optimisation process

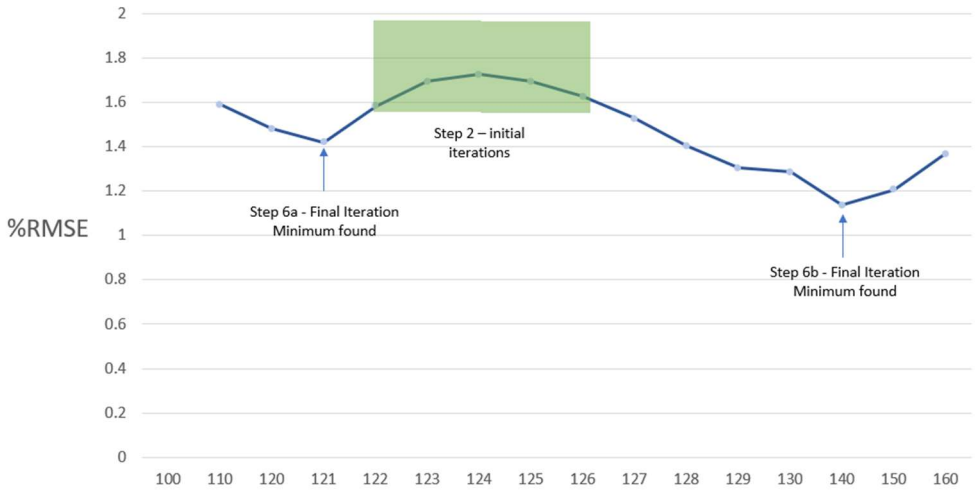

Figure 5.18 Unique situation - both negative

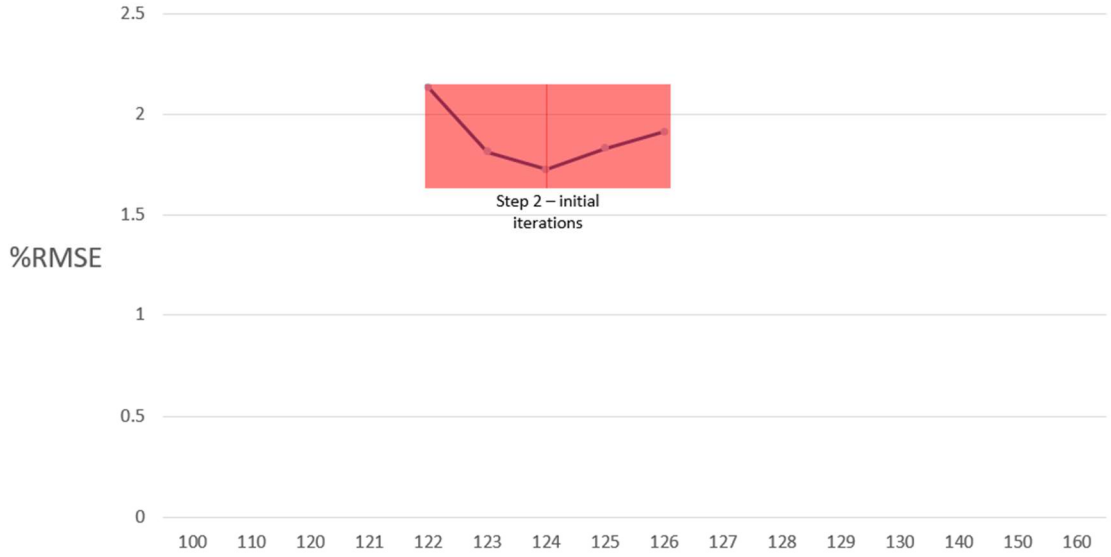

Figure 5.19 Unique situation - both positive

5.2 Multisite Model Proposals

So far, the dataset at each site has only been looked at on a site-by-site basis. This section aims to look at the datasets on a more macroscopic scale and at any relationship between the datasets from multiple sites. From section 3.1.1 There is some evidence that the characteristics of the dataset are very localised in terms of indoor air quality. As such, we will explore a few methods for applying LSTM to multisite data, including a proposal for a new method of applying LSTM to datasets in the hope of improving predictions.

5.2.1 Description of Multisite Prediction methods

5.2.1.1 Large scale Multivariate

The initial approach is to apply all the data from all the sites into a single multivariate LSTM model. In Figure 5.20, we see how such a model would be structured. In this case, the model would look at all the datapoint. The characteristics of data points will not be considered in the process. As such, the model cannot distinguish which site each data point is from, and in our case, it will only know the total number of parameters it has and not know which parameters come from different sites or locations.

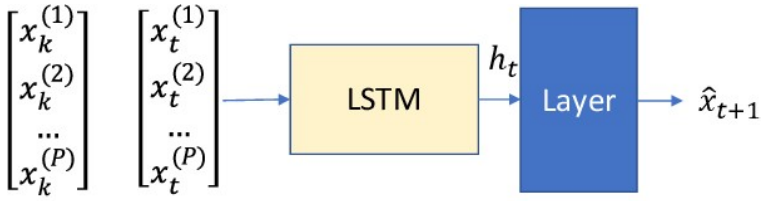

Figure 5.20 Multivariate LSTM

With this method, we are also required to add a synchronisation function between sites to synchronise the number and timestamps of each data point before feeding it into the model. This is due to multivariate LSTM requiring the input data to be synchronous. This synchronisation function causes a minor loss of data in some cases, as at any moment in time if there is a missing data point for any variable, we would have to ignore the datapoint of every other variable at that timestamp. In this application, the synchronisation was performed to the closest minute, and we took one datapoint for each variable at every minute to achieve the synchronisation.

5.2.1.2 Proposed multisite model – Shared hidden layer.

In this approach, we propose to create a variation of Multivariate LSTM that will look at each dataset from each site primarily and individually while incorporating a shared hidden layer between each site that will allow it to potentially gain additional insight from the data from other sites.

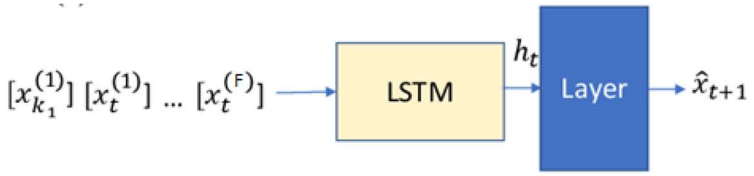

Figure 5.21 Naive single variate LSTM

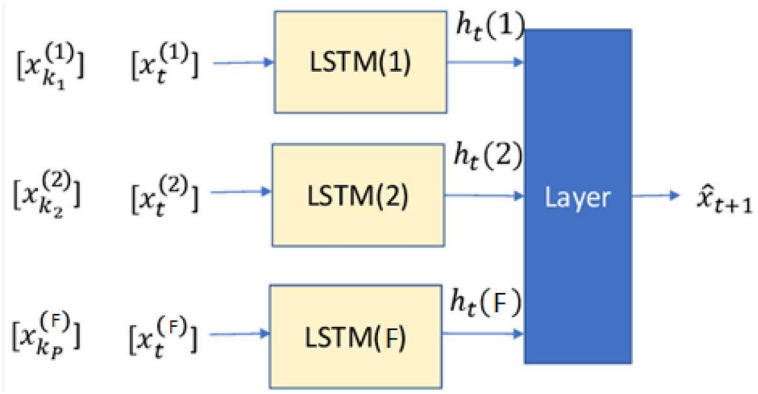

Figure 5.22 Asynchronous single variate

In establishing this proposed method, we first took the structures of a naïve single variate LSTM, shown in Figure 5.21, and a proposed structure for an asynchronous LSTM model, shown in Figure 5.22. We are taking the approach of how an asynchronous model would apply an LSTM Network to each variable individually while incorporating a shared hidden layer. In the proposed method, we have taken the idea of a shared hidden layer from the asynchronous approach and applied it to a multivariate model, as shown in Figure 5.23. However, to achieve such a model, it is impossible to use the existing LSTM equations or even the LSTM equation for the asynchronous approach. This is because the traditional LSTM equation does not consider the hidden layer, while the asynchronous approach equation limits each neural network to a single input variable. As such, we must perform a different approach to achieve this multisite LSTM structure. We will look at three methods of implementing this proposal and their performance.

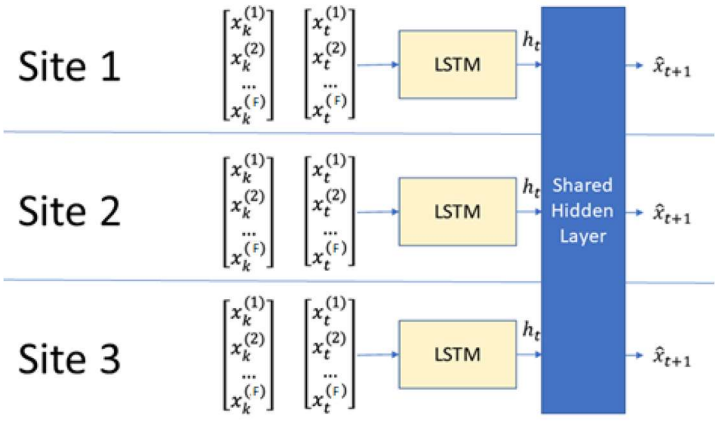

Figure 5.23 Multisite LSTM Proposal

5.2.1.2.1 Proposal A – Shared W and V Hidden state

This approach involves using a normalised version of the two weightage matrices (W and V) across all sites. Using this approach, instead of applying the individual weightage matrices on the LSTM model of each site, we applied an updated weightage matric. The updated weightage matrices are the average of all the sites' weightage matrices from W and V.

5.2.1.2.2 Proposal B - Shared V Hidden state

This proposal involved a similar approach to proposal A. However, instead of normalising both the weightage matrices W and V, we only normalised V and kept W as the individual matric for each site.

5.2.1.2.3 Proposal C – New Shared Hidden state E

With this proposal, we look at implementing a new weightage variable. This weightage variable is calculated using Equation 5.2. This weightage variable is a measure of the error caused by applying any specific hidden state.

New Weightage Factor：

$$E = \frac{1}{N}\sum_{t=1}^{N}(x_{t+1} - \hat{x}_{t+1}) \quad 5.2$$

Based on the original LSTM equations shown earlier in section 2.4, Equation 2.1 to Equation 2.6, and we can then take a simplified version of the LSTM equation, which is highlighted in Equation 5.3

Using this, we can then use the backpropagation through time (BPTT) algorithm to learn the parameters of the LSTM network in order to create and update equations to apply the new weightage value to the equations.

Simplified LSTM Gates：

$$V_t = \begin{bmatrix} O_t \\ i_t \\ u_t \\ f_t \end{bmatrix} \quad 5.3$$

Equation5.4 shows us applying the Error function from Equation 5.3 to simplified LSTM Gates(Equation 5.3) using the BPTT algorithm

BPTT applied to LSTM：

$$\delta V_k = \frac{\partial E}{\partial V_k} \quad 5.4$$

This equation can then be expanded to what is shown in Equation 5.5 if we allow Q=[W,U]

BPTT to LSTM with Q = [W, U] ：

$$\delta Qk = \delta Vk \frac{\partial Vk}{\partial Qk} = \delta Vk[xk\,(1), xk\,(2), \dots, xk\,(P), hk-1] \quad 5.5$$

1

We then decompose the update sequence to Equation 5.6 for each sample of BPTT. From this equation, we can see the effect of a multi-sequence backpropagation update. First, we observe that the same weights are updated as the sum over all timesteps and are not independent. The LSTM combines all information into the hidden state.

Decomposed Update Equation：

$$\delta Q = \sum_{k=1}^{m_1} \delta Q_k[x_k^1, h_{k-1}] + \sum_{k=m_1}^{m_2} \delta Q_k[x_k^1, h_{k-1}] + \sum_{k=m_2}^{m_3} \delta Q_k[x_k^1, h_{k-1}] \quad 5.6$$

In this sequence, we can see that the early stages of training may give a significant error due to the different statistical properties of each variable. We note it may be possible to learn a function where $\mathrm{LSTM}(\mathrm{xk}(3)) \cong \mathrm{LSTM}(\mathrm{xk}(2))$. As such, the series will converge, and we can simplify the update equation to Equation 5.7

Simplified Update Equation： $$\delta Q = \sum_{k=1}^{t} \delta Q_k\left[x_k^p, h_{k-1}\right] \quad 5.7$$

We can then apply to the base LSTM equations. In this application, however, the base LSTM equation will only take inputs from their individual sites, while the update equation will be common across sites and will be this additional shared hidden state we were aiming to create.

5.2.2 Comparing the performance of Multisite Predictions methods

5.2.2.1 Large scale Multivariate

The initial idea of looking at multisite prediction was to use a large Multivariate model and feed this model data from multiple sites, as discussed in section 3.5.1. We tested this using data from2 sites. However, we got some significantly distorted prediction graphs when we tested this approach. Figure 5.24 shows the malformed prediction graphs overlayed with the original data from the two sites used in this test case. It is suspected this is due to the model getting "confused" by the two sets of data that both have their own characteristics due to their location and surrounding circumstances while also having some similar features due to the nature of the pollutant itself.

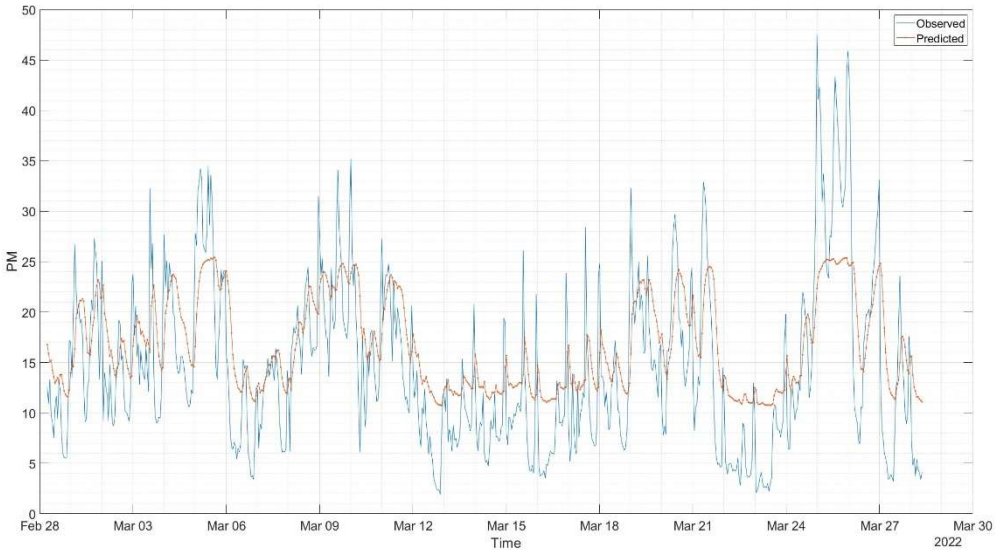

Figure 5.24 Large Scale Multivariate

5.2.2.2 Proposed multisite model – Shared hidden layer

The following approaches look at a proposed modified version of LSTM that incorporates what I would call a shared hidden state. We look at three approaches to achieve this shared hidden state and compare their viability and performance. These approaches are described in section 5.2.1.2, where proposals 1 and 2 involve combining the existing hidden states of the LSTM equation to achieve this shared hidden state, with proposal 1 combining both the W & V hidden states, while proposal 2 involves keeping the W hidden states independent while combining the V hidden state. Proposal 3, instead involves the creation of a new hidden state based on the error function of predictions, described in section 5.2.1.2.

5.2.2.2.1 Site 1 Proposal comparison

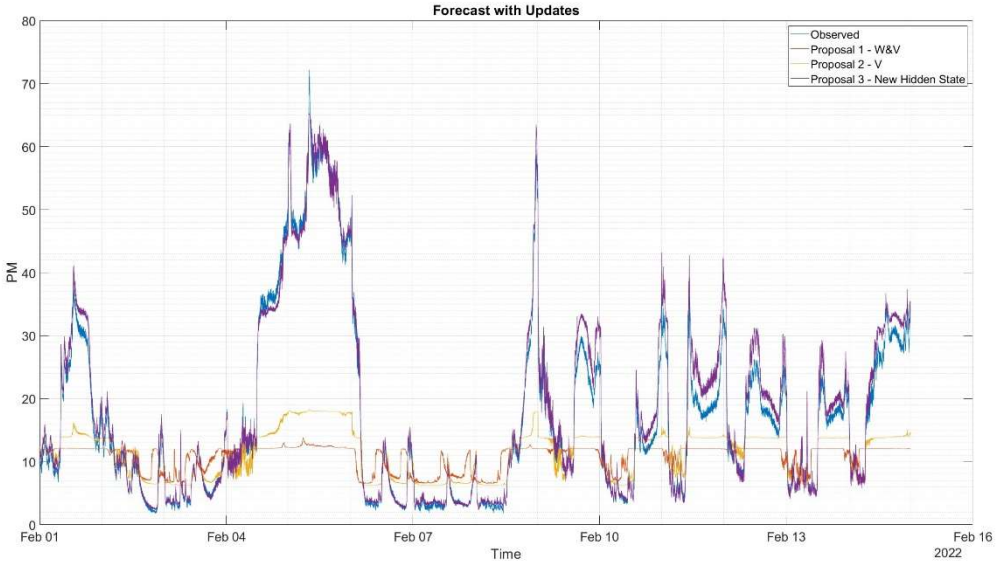

Figure 5.25 Site 1 Proposal Comparison

Figure 5.25 shows the comparison of the prediction on site 1 when using the 3 proposed methods to achieve multisite predictions. We see that with proposals 1 and 2, the amplitude of the peaks and troughs of the predictions are significantly lower than the observed data. In proposal 1 we also see a time shift in the prediction where the predictions are inaccurate on a time basis.

5.2.2.2.2 Site 2 Proposal comparison

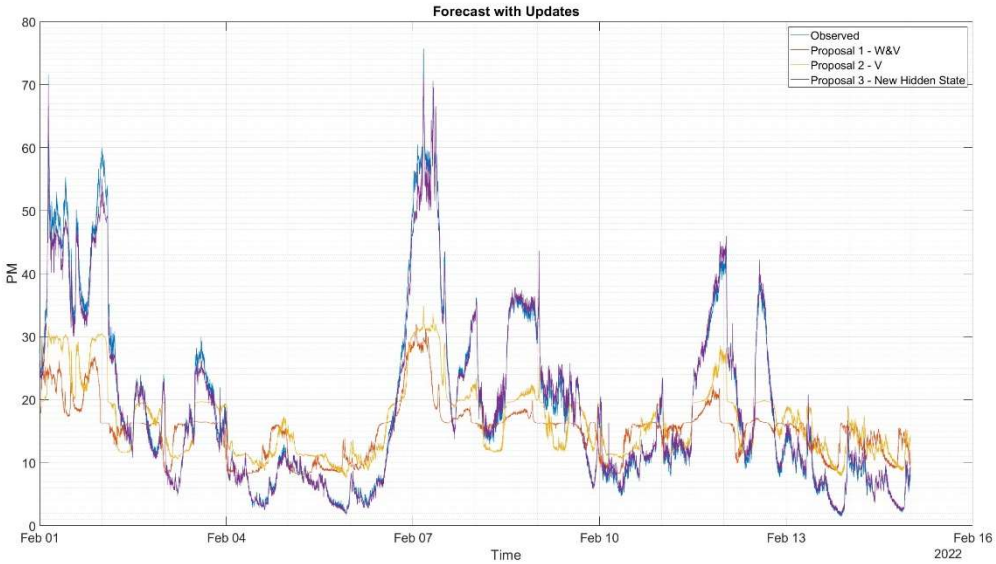

Figure 5.26 Site 2 Proposal Comparison

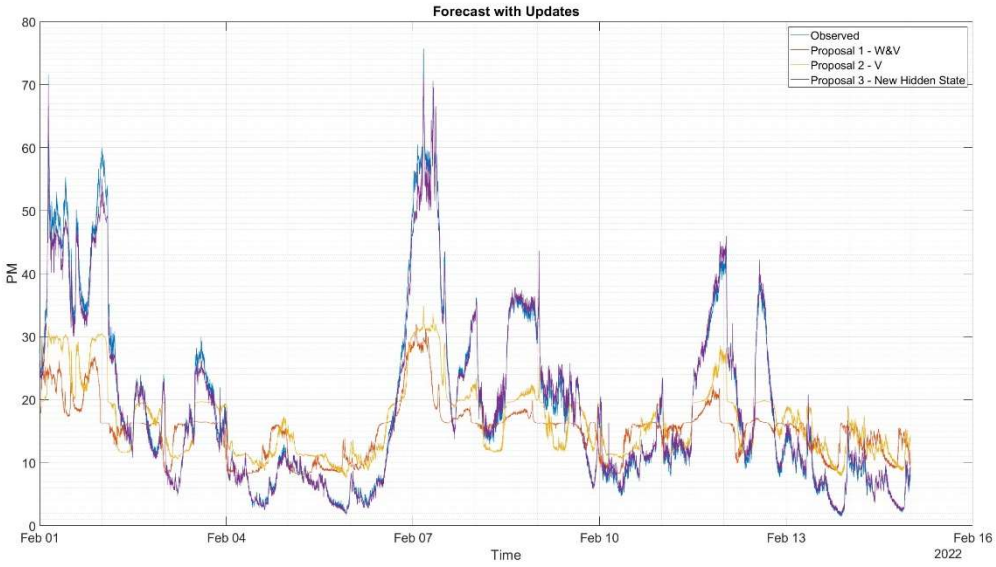

Figure 5.26 shows the comparison of the prediction on site 2 when using the three proposed methods to achieve multisite projections. We see very similar characteristics to the results from Site 1 where with proposals 1 and 2 the amplitude of the predictions are significantly lower than the real data. In proposal 1 we also see the time shift in the prediction where the predictions are inaccurate on a time basis.

We hypothesise that these undesirable characteristics when using proposal 1 and proposal 2 are due to the nature of the W and V Hidden state. The W hidden state is involved in applying a weightage factor to newly input variable values in LSTM. While the V hidden state is involved in the removal of previous and less desirable weightage values of historical data. It is possible that feeding the model data from 2 sites has “confused “its predictions as it is trying to apply these same weightage values to 2 sets of data that could have different characteristics.

To test if the model is getting “confused” in proposals 1 and 2, we tried a special test case where of inputting data from 2 sites into the model, we input the data from a single site into multiple inputs of the model. This resulted in the graphs produced in Figure 5.27. We see that the prediction again has similar accuracy to the original multivariate model. This indicated to us that the model is possibly getting falsely trained by 2 sets of data that are of similar characteristics but have different patterns. i.e. The PM data from site 1 and site 2 would have similar spike characteristics, but they interact with the other variables differently due to the different characteristics of the sites.

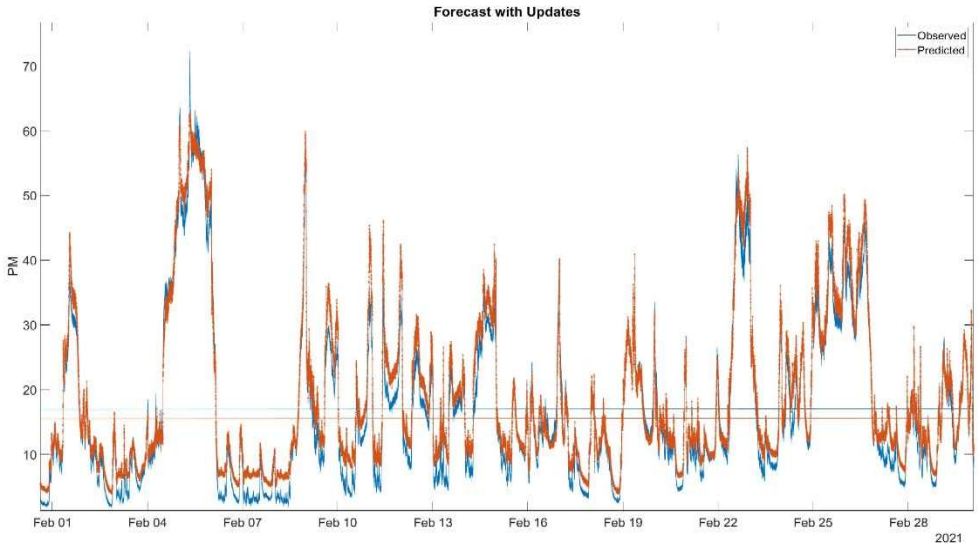

Figure 5.27 Proposal 1 and 2 Special Test Case

5.2.2.2.3 Proposal C – New Shared Hidden State E

Proposal C involves the creation of a new hidden variable that is incorporated into the LSTM equations through the backpropagation through time method. This method is described in section 5.2.1.2.3

5.2.2.2.3.1 Proposal C Effect on Prediction Duration

Initial testing using this technique showed no significant difference when compared to Single site models when looking at 1-hour prediction durations. However, upon testing a larger prediction duration we noticed that the prediction at larger prediction duration were much higher with this variation of the model.

We also made a comparison with the existing multisite model looked at in section 3.3.2 – Geo-BiLSTM. This model was recreated to the best of our ability based on existing documentation and our data was fed into the model while keeping the other training parameters and hyperparameters of the geo-BiLSTM model the same as our proposed model.

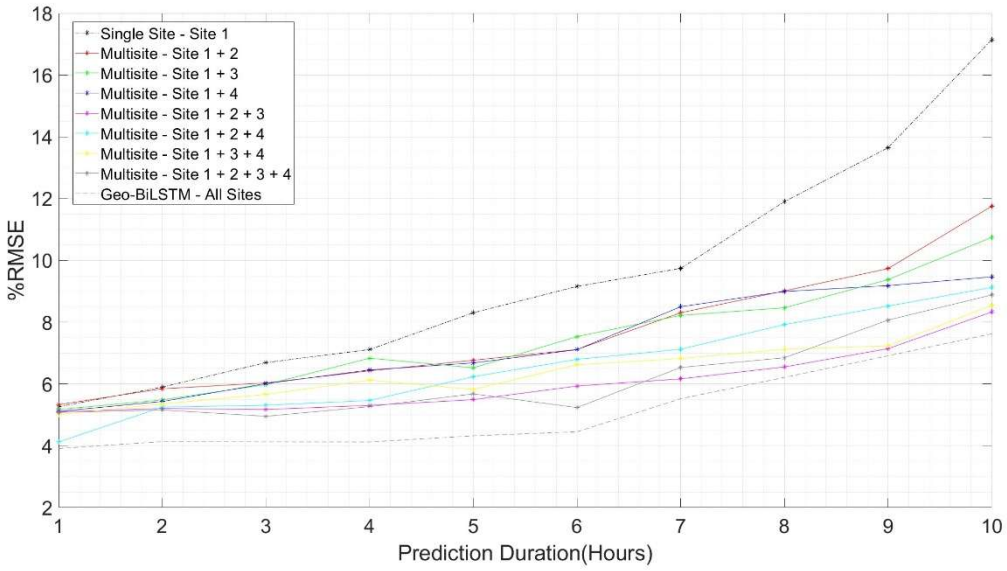

Figure 5.28 Proposal C Prediction duration

Figure 5.28 shows a comparison of the RMSE at different prediction durations with and without using the proposal C multisite variation. With this test, we see the RMSE at lower prediction durations are very similar in both cases. However, as the prediction duration increases, we see that the proposed technique's RMSE remains lower even at large prediction durations of 9 hours. When compared with the existing Geo-BiLSTM, it out performs our proposed model, which is likely due to this model being based on Bi-LSTM instead of a normal LSTM model. We repeated this with multiple site combinations while still looking at Site 1 as the primary site for predictions and measuring the performance of the model, which can be seen in Figure 5.28. In all cases, we see relatively similar results

with some level of randomness but in general, a similar trend amongst all Multisite test cases when compared to Single site.

5.2.2.2.3.2 Proposal C Effect of Training Duration

We then proceed to compare the effects the multisite model has on the training duration of the model.

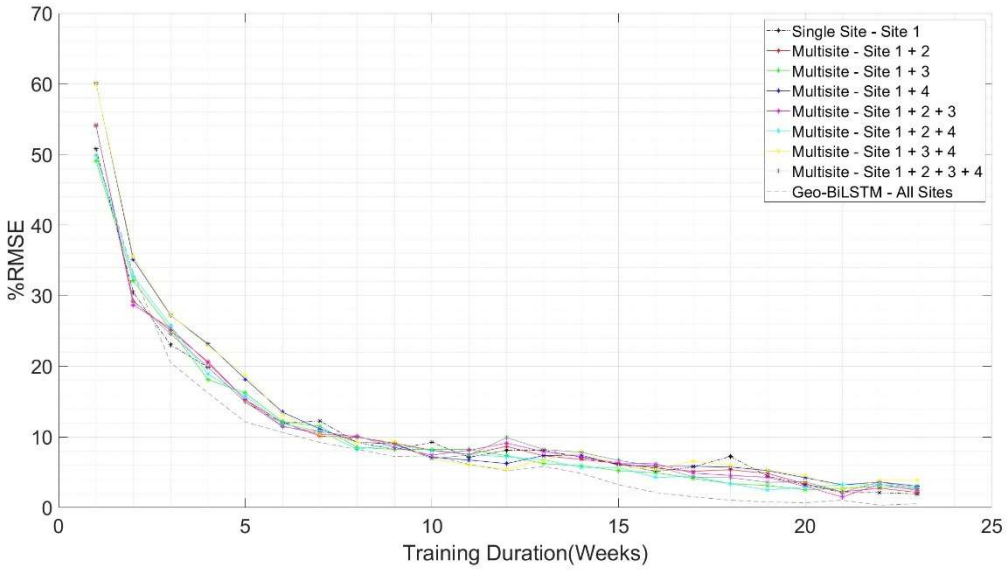

Figure 5.29 Proposal C training Duration

Figure 5.29 shows a comparison of the %RMSE at different training durations with and without using the proposal C multisite variation. With this test, we see marginal differences between the training durations in the case of all 3 site combinations as well as with Geo-Bi LSTM.

5.2.2.2.3.3 Proposal C Effect of Training generations

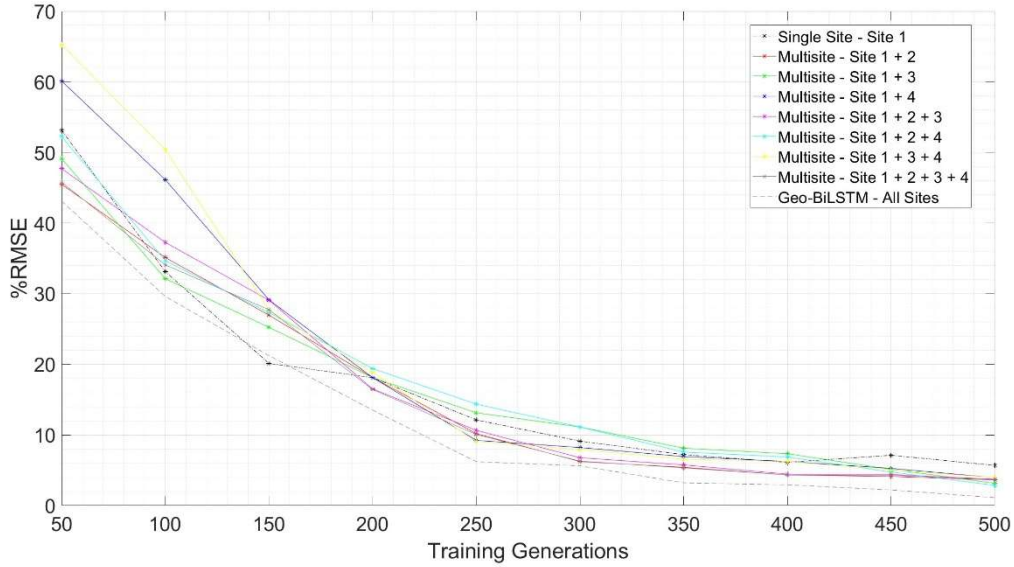

Figure 5.30 Proposal C Training Generations

Figure 5.30 shows a comparison of the %RMSE at different training generations with and without using the proposal C multisite variation. With this test, we see marginal differences between the training durations in the case for all 3 site combinations. We see similar results when compared to what we saw in Section 5.1.1.4, where the 250-300 Training generations point seems to be an ideal stop position as improvements are diminishing. The Existing Geo-Bi-LSTM model also showed very similar performance with our proposed model on al sites. To verify this, we look at plotting the rate of change(gradient) of the %RMSE of the model at each point.

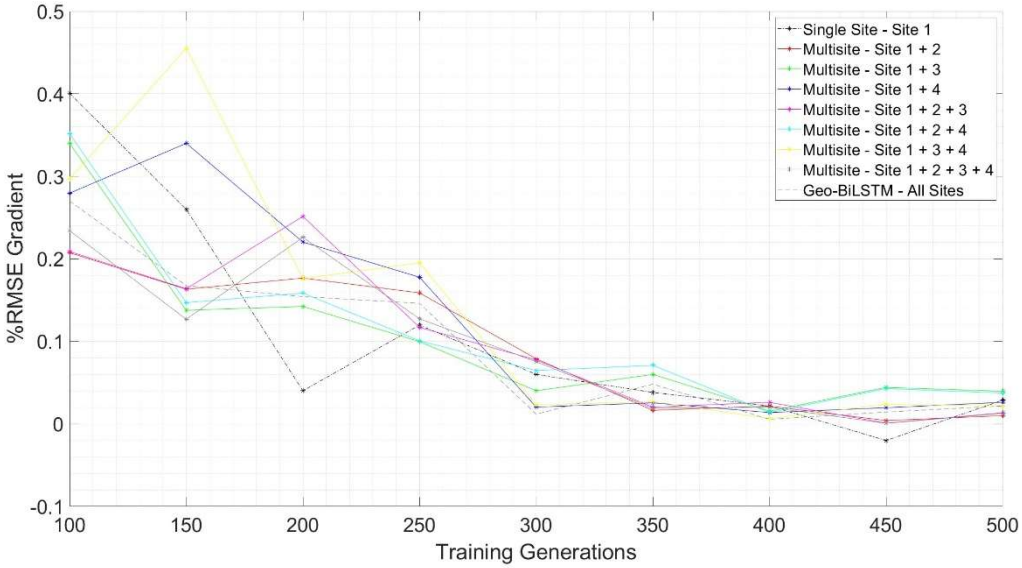

Figure 5.31 Proposal C Training Generations Gradient

Figure 5.31 shows this gradient at each point. We see that past the 300 generation point we consistently get a gradient of approximately 0.05. Also referring to Figure 5.30, the 300-generation mark is also the point where the %RMSE is roughly below the 10% mark.

5.2.2.2.4 Proposal C Staggered Training

While performing these tests we made an observation we noticed slightly varying results when staggering the training of the models. Figure 5.32 and Figure 5.33 show a depiction of applying said offset.

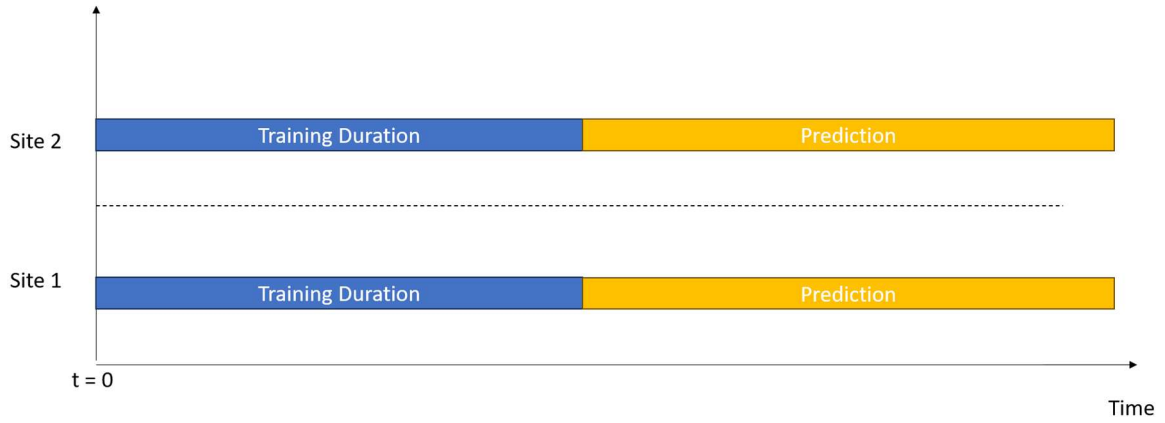

Figure 5.32 Proposal C No Offset

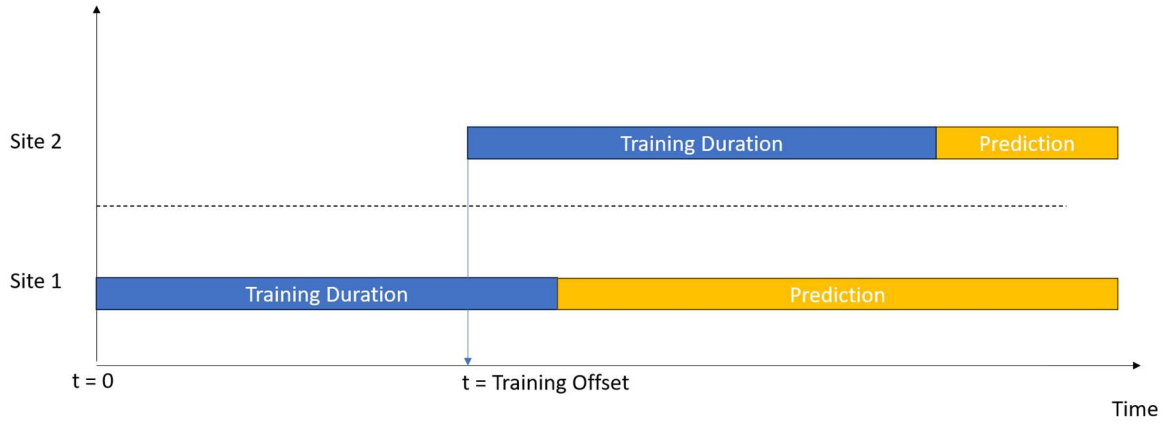

Figure 5.33 Proposal C with offset

Figure 5.32 shows a depiction of using proposal C without the offset. In this situation, we trained the model for both sites simultaneously and thus, the shared hidden state forms from scratch for both sites. Figure 5.33 show us applying the offset to the training of 1 of the 2 sites. In this situation, site 1 is the only site involved the in the initial formation of the shared hidden state. Site 2 on the other case will have access to a developed shared hidden state(from site 1's data) right from the start.

All evaluations using this ofset is done with every site combination using site 1 are a primary site with our proposed methed and using the existing Geo-BiLSTM model on all site but without an offset. It is impossible to incorporate an ofset into the Geo-BiLSTM due to the nature of a normal LSTM model being synchronous. However with our proposed method, due to the model technically being split into multiple LSTM model instead of 1 large model, this allow each site to be asychronnous which is what allows for this training offset to be implemented.

5.2.2.2.4.1 Effects of different sites

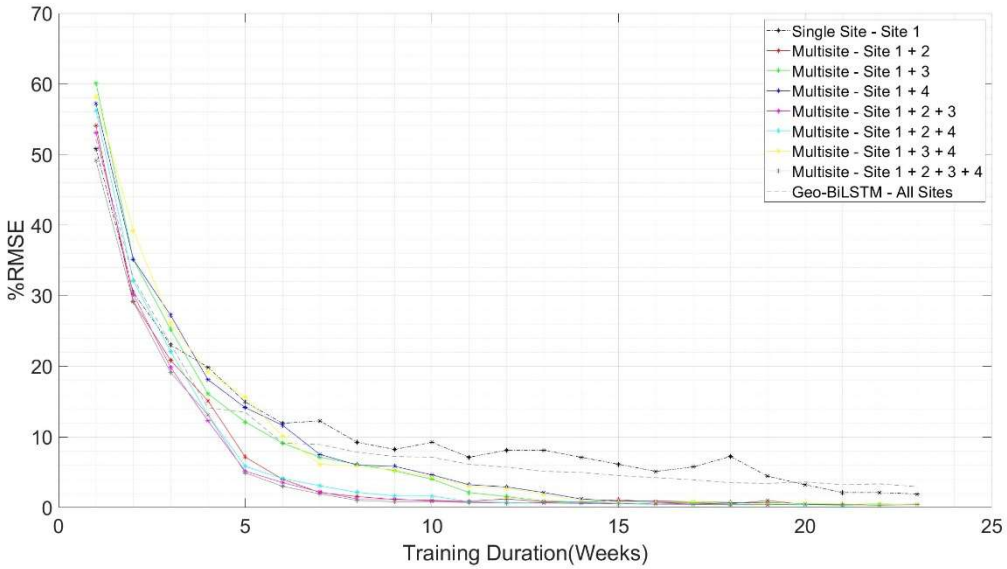

Figure 5.34 Proposal C Staggered Training – Site 1

Figure 5.34 shows the effect of different site combinations on the %RMSE while having a fixed training offset of 16 weeks (3 months). As a benchmark we again used the existing multisite model looked at in section 3.3.2 – Geo-BiLSTM. In Figure 5.34, the largest improvement is provided by the site 1 + site 2 combinations. Incorporating Site 3 and 4 has also shown a minor improvement in the training duration, as seen in the 1+3 and 1+4 combinations. The improvement provided by incorporating different sites seems to vary from site to site. We further see the similar characteristic with the 3 site and 4 site combination, where more significant improvement is observed in any combination that includes site 2. We proceed to look at this relationship by changing the primary site and performing the same test using sites 2, 3 and 4 as the primary site which is used to analyse the performance. When this stagger in introduced we see that our proposed model outperforms the Geo-BiLSTM model, this is like due to the data that our proposed model has incorporated into the shared hidden state before site 1 was incorporated into the model.

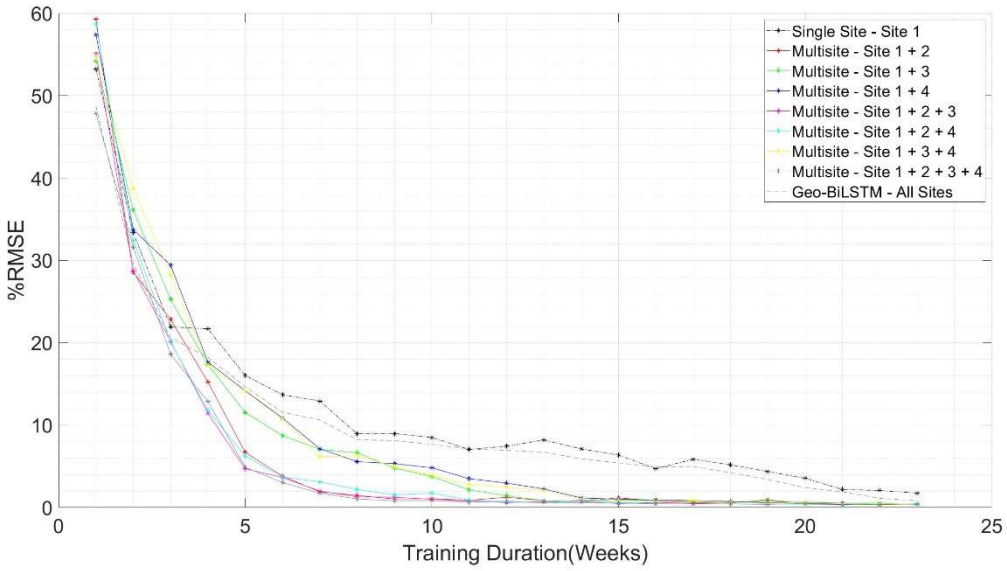

Figure 5.35 Proposal C Staggered Training – Site 2

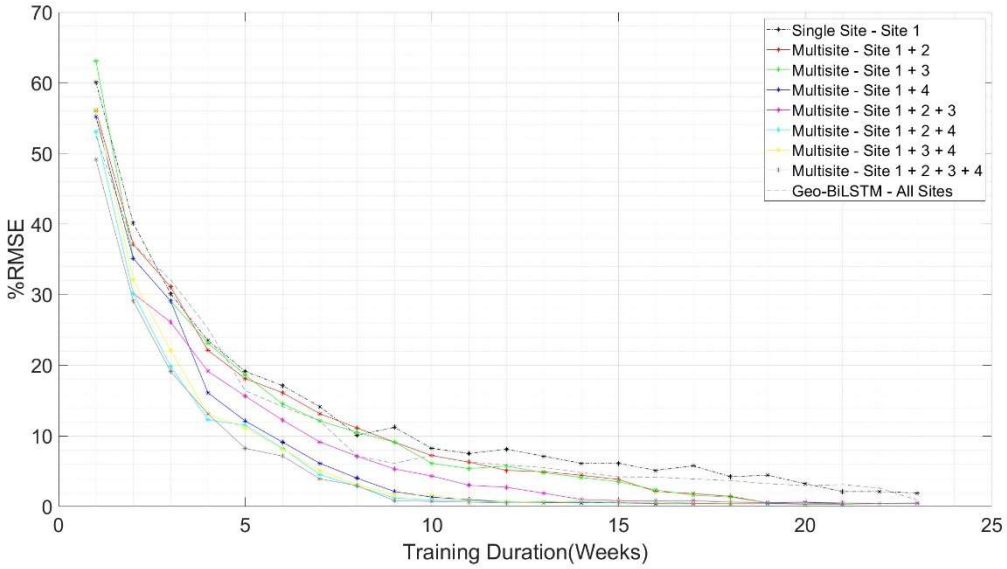

Figure 5.36- Proposal C Staggered Training – Site 3

Figure 5.35 and Figure 5.36 shows a similar analysis of comparing different combinations of site in the multisite model, but this time using site 2 and 3 as the primary site, respectively. The case of Site 2 as the primary site, we see similar characteristics to Site 1 where any combination which includes of site 1 provides a more significant RMSE improvement. This further reinforces our hypothesis that the site characteristics are more significant than the number of sites included in the multisite model. With Site 3 as the primary site, we see combinations including site 4 seem to provide better results compared to combinations including either site 1 or 2.

We suspect different sites provide varying scales of effect on the training duration due to the site's characteristics. Referring to section 4.1.5, we see that Sites 1 and 2 are both apartments a few floors above ground level in urban populated areas. With these sites, we saw they had good synergy with each other providing a significant improvement to the training duration to each other. While Sites 3 and 4 are houses further away from the city and in the same general location. Based on this, we hypothesise that the varying scales of improvement is due to the site characteristics, where the model can more efficiently apply to another site of similar characteristics when compared to applying it to a site of varying characteristics. Our initial four sites seem to confirm this theory, but it would need further testing with more sites to be confirmed.

From this results of our proposed method compared to the normal LSTM and the existing multisite method(Geo-BiLSTM) we see that out proposed method has the benefit of being a synchronous in terms of sites, this means we can incorporate more site into an existing model at a later time. While due to the synchronous nature of LSTM the Geo-BiLSTM model would need to be retrained completely when additional sites are incorporated into it. Furthermore with the stagger introduced our model takes a much shorter time to be trained to a reasonable level. The Geo Bi-LSTM outperforms our proposed model in terms of prediction duration but this likely due to is using Bi-LSTM as a base compared to us using LSTM as a base. Using Bi-LSTm as a base for our proposed method would be something possible to look into the future.

5.3 Chapter Summary

This chapter started by looking at optimising the model using various known methods. We looked at the effects of varying training parameters and how they would affect training accuracy. We also looked at the impact of incorporating multiple data points into the same model. Finally, we proposed a simple method to keep the hyperparameters of LSTM at or close to optimal values. The main aim of the chapter was to optimise the LSTM model to be as suitable as possible for use with indoor air quality predictions. Following this initial optimisation the chapter looks into incorporating data from multiple sites into the modal to achieve a more macroscopic take on the predictions. Based on the proposed multisite methods we demonstrated a novel approach to looking at data from multiple sites into a predictive model. The method incorporated a shared hidden state to link LSTM models from various sites. This shared hidden state was observed to make improvements in the training time and prediction duration of the LSTM prediction.

6 Conclusion

The culmination of this doctoral research manifests a significant stride toward the development of an IoT monitoring solution for smart homes. The creation of a bespoke intelligent IoT system, which incorporates air quality sensing technologies with data from smart home automation systems, stands as a notable contribution to the domain. The system, as validated by the research, exhibits a pronounced capability in forecasting imminent air quality conditions with a high degree of accuracy, courtesy of the employed neural network-based methodologies, particularly the Long Short-Term Memory Neural Network (LSTM).

The devised data collection framework, characterised by a wireless sensor node and an array of strategically deployed sensors within households, proved effective in gathering crucial and reliable data for neural network training. The dynamic predictive model constructed herein, predicated on a continuous influx of real-time air quality data, holds a promising potential in facilitating proactive adjustments to household elements, notably ventilation, thereby ameliorating indoor air quality.

In this study, we investigated various training methods, carefully evaluating their advantages and disadvantages. Ultimately, we decided to adopt the expanding training method due to its optimal balance of high accuracy and manageable training duration. This method integrates a form of real-time error correction, enabling it to sustain commendably high accuracy without necessitating frequent retraining of the model.

Additionally, the exploration of a novel LSTM variant, entailing a shared hidden state, has unfolded a new option for examining interconnected prediction data from multiple locations. This exploration has paved the way for identifying potential correlations between indoor air quality levels across separate sites, which provides benefits in terms of predictions related to indoor air quality.

The study further delved into optimising the LSTM model specifically for IAQ applications, focusing on fine-tuning various training parameters. We concentrated on identifying the minimum viable values for three critical training parameters: training duration, forecast period, and the number of training generations needed to achieve accurate predictions.

The study also explored employing LSTM models to identify and analyse the correlations among various Indoor IAQ factors. Initially, our investigation revealed no straightforward correlations among these IAQ factors. However, the LSTM model exhibited some performance improvements when we shifted to a multivariate approach, hinting at the presence of more complex interrelations. To substantiate these findings, we experimented with various multivariate combinations and cross-referenced other studies, which corroborated that intricate correlations exist between IAQ factors.

The findings and advancements stemming from this study hold promise for the future of IAQ management and invite further exploratory and developmental endeavours in the various applications of this multisite variation beyond IAQ. This versatile approach can potentially deliver benefits in various contexts, opening doors to new opportunities for innovation and progress for various applications.

6.1 Future work

The results of this study demonstrate that our approach enables accurate prediction of indoor air quality (IAQ) across multiple sites. Future directions for this research include expanding the study in two primary ways. First, we aim to incorporate additional data points as well as controllable aspects from home automation systems into the models for each site, potentially enhancing the model's ability to identify sources of poor air quality. Second, we intend to increase the number of sites included in the multisite model to explore the effects of a broader dataset.

6.1.1 Multisite – additional sites

This study was limited to five sites due to constraints in obtaining consent for data collection at additional locations. Moving forward, we plan to expand the model by incorporating more sites to evaluate whether the findings from the initial five locations remain consistent as the sample size increases. Specifically, we aim to assess the improvements in training durations observed with the initial sites, examining the extent of further gains from additional sites and the associated computational costs.

The study also plans to test modifying the Multisite model into a GRU model instead of LSTM in hope to combat the increasing computational cost of incorporating more sites. The study also plans to look into modifying the proposed method to use Bi-LSTM over LSTM to see how it would benefit the model.

6.1.2 Additional datapoints & home automation linkage

To advance this study, the first objective is to investigate the integration of additional data points from home and building automation systems into the model. The purpose of this approach is to assess whether the model can identify sources or causes of poor indoor air quality in households. The underlying hypothesis is that, within the indoor environment, drops in air quality are often caused by controllable factors within the household. Therefore, providing the model with more comprehensive information on household conditions may allow it to “learn” which variables are linked to these dips in air quality.

Once the further datapoints are incorporated into the model, it is planned to first see if there is an effect on the performance of the model, including training time, prediction duration, prediction accuracy and computational cost. In terms of computation cost, it is almost certain that it will increase due to the nature of machine learning. The study then investigates incorporating some other ML techniques into the model, with the aim of identifying the factors that have an effect on the air quality. The initial thought is incorporating something like Multi-layer Feed Forward (MLFF) into the model, as MLFF has been seen to be effective at classifying and categorising types of air pollutants and their sources. [52]

The hope for this expanded model would be to develop a system continues to predict the IAQ while also automatically identifying variables and factors that have an effect on the air quality. Using this information the study hope to use this knowledge to manipulate controllable factors through the building automation system, allowing for the creation of a system that not just predicts negative spikes in air quality but also identifies factors that are related to these spikes and in turn automatically perform actions to mitigate or reduce them.

7 References

[1] 'Indoor air quality — European Environment Agency'. Accessed: Aug. 28, 2020. [Online]. Available: https://www.eea.europa.eu/signals/signals-2013/articles/indoor-air-quality

[2] A. P. Jones, 'Indoor air quality and health', Dec. 01, 1999, *Pergamon*. doi: 10.1016/S1352-2310(99)00272-1.

[3] A. Cincinelli and T. Martellini, 'Indoor air quality and health', Nov. 01, 2017, *MDPI AG*. doi: 10.3390/ijerph14111286.

[4] S. Vilčeková, I. Z. Apostoloski, Ľ. Mečiarová, E. K. Burdová, and J. Kiseľák, 'Investigation of indoor air quality in houses of Macedonia', *Int J Environ Res Public Health*, vol. 14, no. 1, Jan. 2017, doi: 10.3390/ijerph14010037.

[5] V. Van Tran, D. Park, and Y. C. Lee, 'Indoor air pollution, related human diseases, and recent trends in the control and improvement of indoor air quality', Apr. 02, 2020, *MDPI AG*. doi: 10.3390/ijerph17082927.

[6] S. Kephalopoulos, E. Commission, D. Kotzias, T. Arvanitis, P. P. C. S. A, and M. Jantunen, 'THE INDEX-PM PROJECT : HEALTH RISKS FROM EXPOSURE TO INDOOR PARTICULATE MATTER', no. May 2014, 2012, doi: 10.13140/2.1.1052.7688.

[7] S. Kubba and S. Kubba, *Chapter Seven – Indoor Environmental Quality*. Butterworth-Heinemann, 2017. doi: 10.1016/B978-0-12-810433-0.00007-1.

[8] D. Mintz, 'Technical Assistance Document for the Reporting of Daily Air Quality - the Air Quality Index (AQI)', 2016.

[9] G. de Gennaro, G. Farella, A. Marzocca, A. Mazzone, and M. Tutino, 'Indoor and outdoor monitoring of volatile organic compounds in school buildings: Indicators based on health risk assessment to single out critical issues', *Int J Environ Res Public Health*, vol. 10, no. 12, pp. 6273–6291, Nov. 2013, doi: 10.3390/ijerph10126273.

[10] 'AQM Data Sheet'.

[11] H. Chojer, P. T. B. S. Branco, F. G. Martins, M. C. M. Alvim-Ferraz, and S. I. V. Sousa, 'Development of low-cost indoor air quality monitoring devices: Recent advancements', Jul. 20, 2020, *Elsevier B.V.* doi: 10.1016/j.scitotenv.2020.138385.

[12] A. Tiele, S. Esfahani, and J. Covington, 'Design and development of a low-cost, portable monitoring device for indoor environment quality', *J Sens*, vol. 2018, 2018, doi: 10.1155/2018/5353816.

[13] F. Europe, 'Fluke 975 AirMeter specifications Ordering Information Optional accessories ToolPak™ Magnetic Meter Hanging Kit Fluke-975CK AirMeter Calibration Kit Fluke-975VP AirMeter Air Velocity Probe Fluke 975V pictured'.

[14] '(No Title)'. Accessed: Oct. 02, 2020. [Online]. Available: https://br.omega.com/omegaFiles/green/pdf/HHAQ-107.pdf

[15] S. Modbus, ‘PM2 . 5 Particle Counter’, pp. 1–14.

[16] J. Namieśnik, T. Górecki, B. Kozdroń-Zabiega ła, and J. Łukasiak, ‘Indoor air quality (IAQ), pollutants, their sources and concentration levels’, *Build Environ*, vol. 27, no. 3, pp. 339–356, Jul. 1992, doi: 10.1016/0360-1323(92)90034-M.

[17] ‘Common Indoor Air Pollutants’. Accessed: Jul. 08, 2020. [Online]. Available: https://www.iaq.gov.hk/en/1248/common-iaq-pollutants.aspx

[18] Y. F. Xing, Y. H. Xu, M. H. Shi, and Y. X. Lian, ‘The impact of PM2.5 on the human respiratory system’, 2016, *Pioneer Bioscience Publishing*. doi: 10.3978/j.issn.2072-1439.2016.01.19.

[19] Comeap, ‘Cardiovascular Disease and Air Pollution A report by the Committee on the Medical Effects of Air Pollutants’.

[20] M. Lanthier-Veilleux, G. Baron, and M. Généreux, ‘Respiratory diseases in university students associated with exposure to residential dampness or mold’, *Int J Environ Res Public Health*, vol. 13, no. 11, Nov. 2016, doi: 10.3390/ijerph13111154.

[21] ‘Sources and Effects of PM2.5’. Accessed: Sep. 18, 2020. [Online]. Available: https://laqm.defra.gov.uk/public-health/pm25.html

[22] Y. F. Xing, Y. H. Xu, M. H. Shi, and Y. X. Lian, ‘The impact of PM2.5 on the human respiratory system’, 2016, *Pioneer Bioscience Publishing*. doi: 10.3978/j.issn.2072-1439.2016.01.19.

[23] ‘Fine Particulate Matter (PM2.5) in the United Kingdom’, 2012.

[24] C. A. Pope, M. Ezzati, and D. W. Dockery, ‘Fine-particulate air pollution and life expectancy in the United States’, *New England Journal of Medicine*, vol. 360, no. 4, pp. 376–386, Jan. 2009, doi: 10.1056/NEJMsa0805646.

[25] *Estimating Local Mortality Burdens associated with Particulate Air Pollution*. 2014.

[26] M. A. Zoran, R. S. Savastru, D. M. Savastru, and M. N. Tautan, ‘Assessing the relationship between surface levels of PM2.5 and PM10 particulate matter impact on COVID-19 in Milan, Italy’, *Science of the Total Environment*, vol. 738, p. 139825, Oct. 2020, doi: 10.1016/j.scitotenv.2020.139825.

[27] ‘Committee on the Medical Effects of Air Pollutants - GOV.UK’. Accessed: Feb. 01, 2023. [Online]. Available: https://www.gov.uk/government/groups/committee-on-the-medical-effects-of-air-pollutants-comeap

[28] E. Brattich *et al.*, ‘How to get the best from low-cost particulate matter sensors: Guidelines and practical recommendations’, *Sensors (Switzerland)*, vol. 20, no. 11, pp. 1–33, Jun. 2020, doi: 10.3390/s20113073.

[29] C. M. Filley, W. Halliday, and B. K. Kleinschmidt-DeMasters, ‘The Effects of Toluene on the Central Nervous System’, 2004, *American Association of Neuropathologists Inc.* doi: 10.1093/jnen/63.1.1.

[30] J. Shuai *et al.*, 'Health risk assessment of volatile organic compounds exposure near Daegu dyeing industrial complex in South Korea', *BMC Public Health*, vol. 18, no. 1, Apr. 2018, doi: 10.1186/s12889-018-5454-1.

[31] O. US EPA, 'Volatile Organic Compounds' Impact on Indoor Air Quality'.

[32] D. Rüffer, F. Hoehne, and J. Bühler, 'New digital metal-oxide (MOx) sensor platform', *Sensors (Switzerland)*, vol. 18, no. 4, Apr. 2018, doi: 10.3390/s18041052.

[33] 'Carbon Dioxide | Wisconsin Department of Health Services'. Accessed: Sep. 25, 2020. [Online]. Available: https://www.dhs.wisconsin.gov/chemical/carbondioxide.htm

[34] 'Carbon Dioxide: Your Environment, Your Health | National Library of Medicine'. Accessed: Sep. 25, 2020. [Online]. Available: https://toxtown.nlm.nih.gov/chemicals-and-contaminants/carbon-dioxide

[35] 'Datasheet Sensirion SCD30 Sensor Module'. Accessed: Oct. 12, 2020. [Online]. Available: www.sensirion.com

[36] L. Fang, 'Impact of temperature and humidity on the perception of indoor air quality', *Indoor Air*, vol. 8, no. 2, pp. 80–90, 1998, doi: 10.1111/j.1600-0668.1998.t01-2-00003.x.

[37] P. Wolkoff, 'Indoor air humidity, air quality, and health – An overview', Apr. 01, 2018, *Elsevier GmbH*. doi: 10.1016/j.ijheh.2018.01.015.

[38] 'IC sensors'. Accessed: Sep. 25, 2020. [Online]. Available: https://www.omega.co.uk/prodinfo/Integrated-Circuit-Sensors.html

[39] 'Humidity Sensor - Types and Working Principle'. Accessed: Sep. 25, 2020. [Online]. Available: https://www.electronicshub.org/humidity-sensor-types-working-principle/

[40] 'NB-IoT vs Zigbee: A detailed comparison'. Accessed: Oct. 08, 2024. [Online]. Available: https://www.narrowband.com/nb-iot-vs-zigbee

[41] 'Wireless Connectivity Options for IoT Applications - Technology Comparison | Bluetooth® Technology Website'. Accessed: Oct. 08, 2024. [Online]. Available: https://www.bluetooth.com/blog/wireless-connectivity-options-for-iot-applications-technology-comparison/

[42] 'What is LSTM? Introduction to Long Short-Term Memory | by Rebeen Hamad | Medium'. Accessed: Oct. 21, 2024. [Online]. Available: https://medium.com/@rebeen.jaff/what-is-lstm-introduction-to-long-short-term-memory-66bd3855b9ce

[43] 'What is LSTM - Long Short Term Memory? - GeeksforGeeks'. Accessed: Oct. 21, 2024. [Online]. Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

[44] 'Understanding LSTM Networks -- colah's blog'. Accessed: Oct. 21, 2024. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[45] ‘Understanding LSTM Networks -- colah’s blog’. Accessed: Aug. 31, 2023. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[46] M. Cameletti, R. Ignaccolo, and S. Bande, ‘Comparing air quality statistical models’, Nov. 2010, Accessed: Oct. 02, 2020. [Online]. Available: http://arxiv.org/abs/1011.1845

[47] C. M. Vong, W. F. Ip, P. K. Wong, and J. Y. Yang, ‘Short-term prediction of air pollution in macau using support vector machines’, *Journal of Control Science and Engineering*, vol. 2012, 2012, doi: 10.1155/2012/518032.

[48] J. Pallant and J. Pallant, ‘Multivariate analysis of variance’, in *SPSS Survival Manual*, 2020, pp. 300–314. doi: 10.4324/9781003117452-26.

[49] M. Kim, B. Sankararao, O. Kang, J. Kim, and C. Yoo, ‘Monitoring and prediction of indoor air quality (IAQ) in subway or metro systems using season dependent models’, *Energy Build*, vol. 46, pp. 48–55, 2011, doi: 10.1016/j.enbuild.2011.10.047.

[50] G. A. Caceres *et al.*, ‘An Application of ARIMA modelling to air pollution concentrations during covid pandemic in Italy A modified ARIMA model for forecasting chemical sales in the USA Comparison of different predictive models and their effectiveness in sunspot number prediction An Application of ARIMA modelling to air pollution concentrations during covid pandemic in Italy’, *J Phys Conf Ser*, vol. 2162, p. 12009, 2022, doi: 10.1088/1742-6596/2162/1/012009.

[51] A. I. Syazwan *et al.*, ‘Analysis of indoor air pollutants checklist using environmetric technique for health risk assessment of sick building complaint in nonindustrial workplace’, *Drug Healthc Patient Saf*, vol. 4, no. 1, pp. 107–126, Sep. 2012, doi: 10.2147/DHPS.S33400.

[52] S. Mad Saad, A. M. Andrew, A. Y. M. Shakaff, A. R. Mohd Saad, A. M. Y. Kamarudin, and A. Zakaria, ‘Classifying sources influencing indoor air quality (IAQ) using artificial neural network (ANN)’, *Sensors (Switzerland)*, vol. 15, no. 5, pp. 11665–11684, May 2015, doi: 10.3390/s150511665.

[53] J. Ahn, D. Shin, K. Kim, and J. Yang, ‘Indoor air quality analysis using deep learning with sensor data’, *Sensors (Switzerland)*, vol. 17, no. 11, Nov. 2017, doi: 10.3390/s17112476.

[54] R. Mumtaz *et al.*, ‘Internet of Things (IoT) Based Indoor Air Quality Sensing and Predictive Analytic—A COVID-19 Perspective’, *Electronics 2021, Vol. 10, Page 184*, vol. 10, no. 2, p. 184, Jan. 2021, doi: 10.3390/ELECTRONICS10020184.

[55] P. K. Sharma *et al.*, ‘IndoAirSense: A framework for indoor air quality estimation and forecasting’, *Atmos Pollut Res*, vol. 12, no. 1, pp. 10–22, Jan. 2021, doi: 10.1016/J.APR.2020.07.027.

[56] E. Hossain, M. A. U. Shariff, M. S. Hossain, and K. Andersson, 'A novel deep learning approach to predict air quality index', *Advances in Intelligent Systems and Computing*, vol. 1309, pp. 367–381, 2021, doi: 10.1007/978-981-33-4673-4_29.

[57] N. Fernandes and J. Gonçalves, 'Multivariate and multi-output indoor air quality prediction using bidirectional LSTM', in *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, May 2023, pp. 1–6. doi: 10.1109/ISDFS58141.2023.10131695.

[58] Y. Wei, J. Jang-Jaccard, W. Xu, F. Sabrina, S. Camtepe, and M. Boulic, 'LSTM-Autoencoder-Based Anomaly Detection for Indoor Air Quality Time-Series Data', *IEEE Sens J*, vol. 23, no. 4, pp. 3787–3800, Feb. 2023, doi: 10.1109/JSEN.2022.3230361.

[59] J. Duan, Y. Gong, J. Luo, and Z. Zhao, 'Air-quality prediction based on the ARIMA-CNN-LSTM combination model optimized by dung beetle optimizer', *Scientific Reports |*, vol. 13, p. 12127, 123AD, doi: 10.1038/s41598-023-36620-4.

[60] S. Miao, M. Gangolells, and B. Tejedor, 'Data-driven model for predicting indoor air quality and thermal comfort levels in naturally ventilated educational buildings using easily accessible data for schools', *Journal of Building Engineering*, vol. 80, p. 108001, Dec. 2023, doi: 10.1016/J.JOBE.2023.108001.

[61] L. T. Wong, K. W. Mui, and T. W. Tsang, 'Updating Indoor Air Quality (IAQ) Assessment Screening Levels with Machine Learning Models', *Int J Environ Res Public Health*, vol. 19, no. 9, p. 5724, May 2022, doi: 10.3390/IJERPH19095724.

[62] J. Loy-Benitez, P. Vilela, Q. Li, and C. Yoo, 'Sequential prediction of quantitative health risk assessment for the fine particulate matter in an underground facility using deep recurrent neural networks', *Ecotoxicol Environ Saf*, vol. 169, pp. 316–324, Mar. 2019, doi: 10.1016/j.ecoenv.2018.11.024.

[63] R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, and F. Li, 'Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering', *Expert Syst Appl*, vol. 169, p. 114513, May 2021, doi: 10.1016/J.ESWA.2020.114513.

[64] T. Jia, G. Cheng, Z. Chen, J. Yang, and Y. Li, 'Forecasting urban air pollution using multi-site spatiotemporal data fusion method (Geo-BiLSTMA)', *Atmos Pollut Res*, vol. 15, no. 6, p. 102107, Jun. 2024, doi: 10.1016/J.APR.2024.102107.

[65] 'NUCLEO-L432KC - STM32 Nucleo-32 development board with STM32L432KC MCU, supports Arduino nano connectivity - STMicroelectronics'. Accessed: Nov. 02, 2022. [Online]. Available: https://www.st.com/en/evaluation-tools/nucleo-l432kc.html#

[66] 'NUCLEO-XXXXKX', 2019. Accessed: Oct. 02, 2020. [Online]. Available: www.st.com

[67] SHARP, 'GP2Y1010AU0F DATASHEET', 2006.

[68] Bosch, 'BME280-Data sheet', 2018.

[69] ‘Humidity Sensor BME280 | Bosch Sensortec’. Accessed: Nov. 29, 2022. [Online]. Available: https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/#documents

[70] ‘CCS811 - ScioSense.’ Accessed: Nov. 29, 2022. [Online]. Available: https://www.sciosense.com/products/environmental-sensors/ccs811/

[71] ams AG, ‘ams Datasheet CCS811 Ultra-Low Power Digital Gas Sensor for Monitoring Indoor Air Quality’.

[72] ‘Lithium ion Rechargeable battery Cell Type NCR18650B Specifications 2G23X0KYKU’.

[73] ‘NanJing Top Power ASIC Corp. TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8 DESCRIPTION’.

[74] ‘Raspberry Pi Datasheets’. Accessed: Nov. 02, 2022. [Online]. Available: https://datasheets.raspberrypi.com/

[75] ‘Documentation - MariaDB.org’. Accessed: Nov. 02, 2022. [Online]. Available: https://mariadb.org/documentation/

[76] ‘Multi-Source Replication - MariaDB Knowledge Base’. Accessed: Nov. 28, 2022. [Online]. Available: https://mariadb.com/kb/en/multi-source-replication/